LO SVILUPPO DI UN'AVVENTURA

Eccoci in un mondo di fantasia, creato da noi stessi per stupire e anche un po' per sconvolgere gli amici. Diamo uno sguardo al mondo e alla storia dei giochi di avventura

Per chi desideri un po' di tregua dai giochi a base di velocità e mira, l'alternativa è il gioco d'avventura, un genere di gioco grazie al quale ci si immerge totalmente in un mondo di fantasia creato dallo stesso programmatore. Esercitando intelligenza, saggezza e conoscenza di fatti ed eventi insoliti, si conduce un viaggio alla ricerca dell'obiettivo creato dall'ideatore.

Nelle successive sezioni di Giochi al Computer, verrà spiegato come creare da zero un proprio gioco d'avventura. Per adesso, ecco un'introduzione a questo interessante genere.

LA STORIA DELLE "ADVENTURE"

L'idea di scrivere giochi d'avventura proviene dalla popolarità di giochi classici (non da computer) quali il "Dungeons e Dragons" e dal desiderio di fare con il computer qualcosa di più interessante della semplice elaborazione dati. In "Dungeons e Dragons" i giocatori entrano nei panni di alcuni personaggi e si affrontano l'un l'altro in un mondo, Dungeon, creato dal Signore di Dungeon. Questo ruolo, nei giochi d'avventura computerizzati, è in parte svolto del programmatore che, come il Signore di Dungeon, crea un proprio mondo. Il giocatore, d'altra parte, assume un ruolo simile a quello di uno dei personaggi.

Tuttavia, nei giochi d'avventura al computer, i giocatori di solito non possono scegliere il tipo di personaggio: è il computer a fare questa scelta. Nelle versioni più sofisticate, si può anche scegliere l'equipaggiamento da portarsi appresso, prima di partire verso l'obiettivo, ma partecipazione e verosimiglianza dell'avventura dipendono tutte dall'ideatore. La prima avventura fu scritta in FORTRAN, anziché in BASIC, su una grande unità di elaborazione. Il programma occupava più di 300 Kbyte di memoria, molto di più di quella di un microcomputer.

La storia vera e propria cominciò però nel 1978, quando Scott Adams trasferì alcune idee su un micro TRS 80, dimostrando la possibilità di scrivere un'avventura soddisfacente in uno spazio di memoria molto minore.

I temi adottati da Adams per i suoi giochi (Adventureland, Covo dei Pirati, La casa dei misteri e Il Conte), sono stati da allora una fonte inesauribile di ispirazione per gli ideatori di giochi.

TIPI DI AVVENTURA

Nel gioco originale, ma anche in quelli di Adams, sullo schermo vengono visualizzati soltanto dei testi. Questo tipo di gioco, esclusivamente testuale, è ancora molto apprezzato e, sostengono alcuni, costituisce la vera avventura.

In effetti, nelle avventure puramente testuali, la trama esiste soltanto nella mente del giocatore che, se il programma è buono, sarà del tutto preso dallo svolgersi degli eventi.

Solo una grafica molto sofisticata può competere con la nostra immaginazione. Per esempio, si può immaginare un orco molto più feroce di quanto si riesca a disegnare con la migliore grafica.

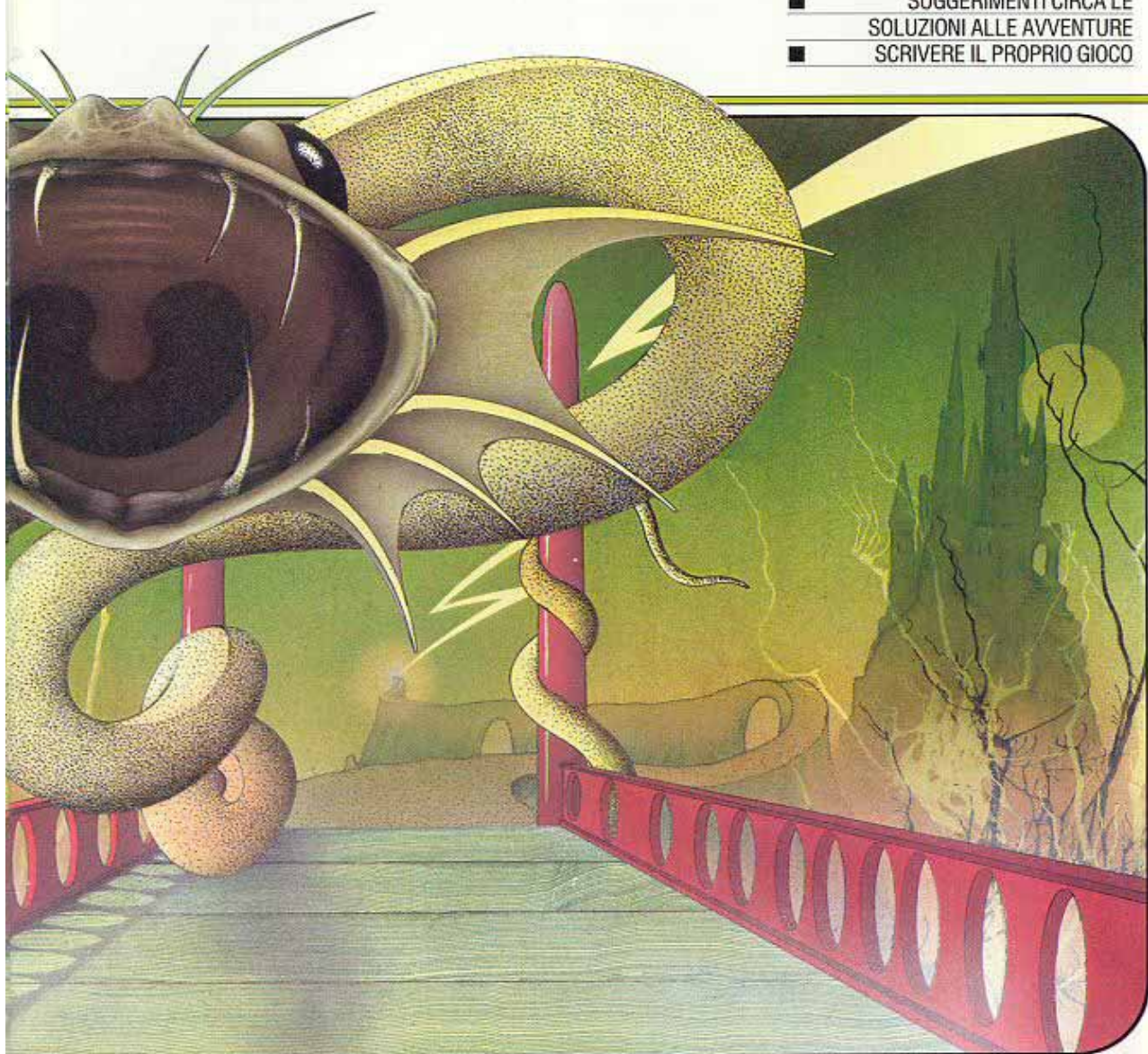
Una considerazione più concreta contro l'uso della grafica è che la gran quantità di memoria ad essa dedicata potrebbe invece servire per allargare la portata del gioco. Inoltre, il ritmo dell'avventura si infiacchisce, se il giocatore deve attendere ogni volta il formarsi del disegno.

Certe avventure assegnano un punteggio al raggiungimento di determinate fasi del gioco, in modo da sapere, a qualsiasi livello si venga eventualmente uccisi, come ci si è comportati. Altri, invece, esprimono un giudizio del tipo 'buffone o esperto'. Un tipo di avventura diametralmente opposto non fornisce intenzionalmente alcuna indicazione su come si stia procedendo, né su quanto si è vicini alla

Cosa fare se si resta bloccati in mezzo a un'avventura?

Prima di arrenderci, riproviamo dopo un po' di tempo: può venire in mente un'altra idea o strategia. Ma se si è proprio bloccati, le avventure in commercio forniscono spesso fascicoletti con le soluzioni, o danno modo di richiederli per posta.

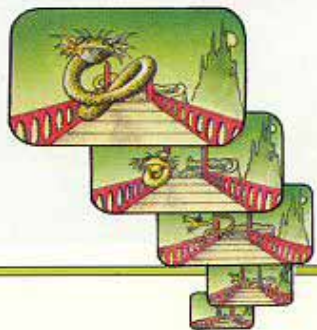
Un altro canale da seguire per un aiuto sono le rubriche di alcune riviste specializzate.

- COS'È UN GIOCO D'AVVENTURA
- COME SI GIOCA
- SUGGERIMENTI CIRCA LE SOLUZIONI ALLE AVVENTURE
- SCRIVERE IL PROPRIO GIOCO

meta: la soddisfazione finale proviene dalla risoluzione di una lunga serie di rompicapi, che ci avvicinano gradualmente al termine della ricerca e all'obiettivo finale.

COME GIOCARE

Di norma, quando si inizia un'avventura, viene fornita una descrizione sommaria sul mondo in cui ci si addentra: se cioè si tratta di un luogo inesplorato o di un pianeta sperduto oppure di un mondo del tutto fantastico.

Il gioco può ambientarsi nel passato, nel presente o nel futuro... o anche in tutti e tre i momenti. In genere, viene data qualche indicazione iniziale (chi comanda nel mondo, qual'è il ruolo del giocatore, chi sono gli amici e chi i nemici e via dicendo) e, dato più importante, cosa occorre fare per riuscire nell'avventura e vin-

cere la partita. Le istruzioni vanno lette attentamente, per non perdere importanti informazioni. Appare quindi la descrizione del primo ambiente, del tipo:

SEI IN PIEDI ACCANTO AD UN GROSSO CALDERONE PIENO FINO ALL'ORLO DI UN RIBOLLENTE FLUIDO VERDE. NELL'ARIA C'È UNA PUZZA INFERNALE. PER TERRA C'È UN CUCCHIAIO.
PUOI MUOVERTI A: EST
OVEST
NORD
E ADESSO?

Ora si deve decidere cosa fare. Si intende usare il cucchiaio per mescolare il liquido o addirittura assaggiarlo? O lo si lascia stare? O si va alla ricerca di una bottiglia o di un contenitore per portarsi dietro un po' di questo misterioso fluido?

Se si decide di usare il mestolo, occorre rispondere qualcosa come:

PRENDI IL CUCCHIAIO

al che il computer risponderà OK o NON ANCORA! oppure con un altro messaggio diverso.

Ad ogni fase del gioco è necessario comunicare al computer ciò che si intende fare. Il come dipende dal gioco: alcuni giochi attendono le istruzioni sotto forma di un verbo seguito da un nome (PRENDI IL MESTOLO, STRANGOLA L'ELEFANTE ecc.)

Giochi più raffinati accettano frasi intere, ma ciò avviene raramente. In questo tipo di gioco si può rispondere qualcosa come UCCIDO L'INSETTO CALPESTANDOLO MENTRE CANTO 'SAPORE DI SALE'. Un programma con istruzioni così complicate sarebbe di certo molto complesso e fuori dalla portata di un principiante.

Quasi tutti questi giochi accettano (o si aspettano), abbreviazioni delle parole, come N per NORD, che sveltiscono il gioco e risparmiano spazio in memoria.

Non ci sono solo le direzioni N, S, E, O; si può dover andare in Su o GIÙ, o anche a NE, SE, SO e NO. Se il gioco non avverte sulle direzioni consentite, occorre provare tutte le possibilità. Lo schema abituale del percorso di un'avventura è basato su una griglia di possibili ambienti, generalmente disposti in un quadrato. Cosa debbano rappresentare questi ambienti sta alla fantasia del programmatore: dalle stanze di un castello, alle gallerie di una profonda miniera. Il confine tra gli ambienti può essere qualcosa di evidente, come una porta o un sentiero o delle scale, oppure qualcosa di più incerto, come un fiume da attraversare a nuoto.

LA SOLUZIONE DELLE AVVENTURE

Normalmente, esiste una sola soluzione per un'avventura; ad esempio: raccogliere tutto il tesoro e riportarlo al sicuro, o eliminare il rappresentante del Male e sfuggire illesi.

Il resto è una sequenza fissa di problemi da risolvere. Le probabilità sono tali che i tentativi saranno molti prima che l'avventura finisca. Infatti, le avventure più belle sono quelle che richiedono giorni, settimane, o addirittura mesi di fatica! Ci sono però alcune regole e suggerimenti per aiutare a risolvere quasi tutte le avventure in meno tempo.

Quasi senza eccezioni, gli oggetti che si trovano nelle avventure servono a qualcosa. Sarebbe uno spreco di memoria se il programmatore lasciasse in giro troppi diversivi, anche se alcuni oggetti saranno armi a doppio taglio. Per esempio, può essere necessario portarsi dietro un sacco di monete d'oro per pagare il pedaggio sul ponte, ma se si deve invece passare il fiume a nuoto, il peso ci farà annegare. In genere, conviene prendere quanti più oggetti possibile, ma in qualche occasione si vedrà che non si possono tenere tutti. Molti di essi servono solo una volta nell'avventura, salvo forse una spada, ad esempio, che servirà più volte per eliminare i cattivi. Se c'è un limite al numero di oggetti da portare con sé, ricordiamo che spesso è prudente scartare gli oggetti dopo l'uso.

È *sempre* bene tracciare una mappa, segnandovi i nomi delle stanze e le cose interessanti in ognuna, oltre agli oggetti che vi si trovano, le entrate, le uscite e le loro direzioni.

Questa mappa ci risparmierà un sacco di tempo, evitando di tornare sui propri passi, cosa che accade spesso durante il gioco. Se ci sono oggetti da abbandonare, perché non si può portare tutto con sé, si segni la loro posizione sulla mappa. E, altrettanto importante, una mappa dà la garanzia di esplorare tutta l'avventura senza per questo ricadere per l'ennesima volta nelle "solite" sabbie mobili.

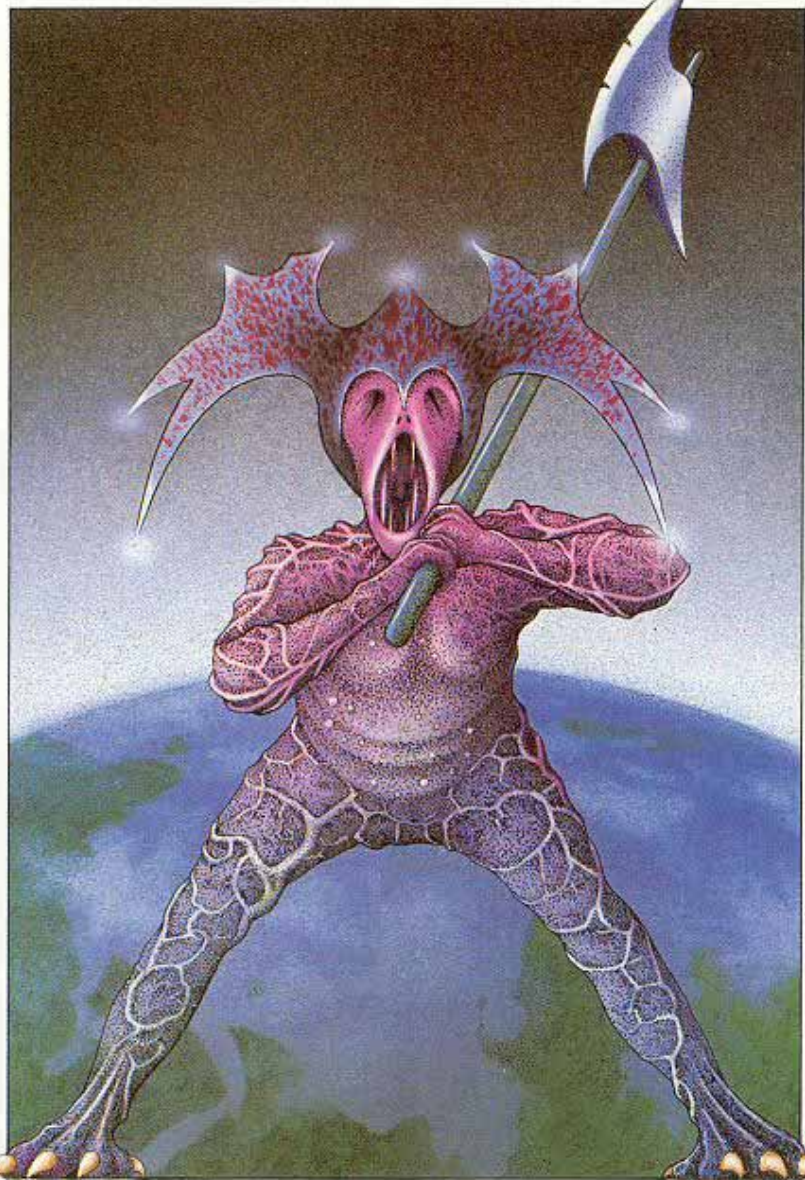
La maggior parte dei giochi prevede la visualizzazione di un inventario degli oggetti che si hanno con sé e per esaminarlo di fronte a un ostacolo basta scrivere INVENTARIO, INVE o solo I, a seconda dell'avventura.

Allo stesso modo, alcune avventure permettono di chiedere aiuto e la formula dipende anche stavolta dal gioco. Può esserci o non esserci un aiuto, ma alla peggio ci si troverà di fronte a "NESSUN AIUTO QUI!". Alcuni giochi seguono da vicino la trama di un particolare libro e in questo caso studiarsi il libro in questione è senz'altro una buona idea. A volte sono state riprese solo piccole sezioni o idee e, se ci sembra di riconoscere qualcosa, proviamo a ricercarla sul libro.

Analogamente, se un personaggio ripugnante che brandisce una pesante mannaia ci sbarra la strada chiedendoci quale sia il diametro della Terra, non cerchiamo di tirare a indovinare, ma corriamo a consultare un'enciclopedia!

Un'altra trappola molto usata sono i quiz con giochi di parole: attenzione, non sempre sono facili come sembra. Teniamoci quindi vicino un vocabolario o un dizionario dei sinonimi per risolvere doppi sensi e frasi ambigue: il programmatore potrebbe aver introdotto LUSTRARE, ma non STROFINARE, ad esempio.

Un ultimo avvertimento: se l'avventura che si sta giocando permette di memorizzare su nastro o su disco una particolare fase del gioco, per poterlo giocare a più riprese, è bene sfruttare l'occasione prima di affrontare una difficoltà. Se andasse male, si può sempre riprende-

re da dove si era rimasti e tentare così più volte l'assalto a un drago o il passaggio di un ponte pericolante.

SCRIVERE AVVENTURE

Scrivere avventure è un buon modo per esercitarsi in BASIC: vengono utilizzati quasi tutti gli aspetti importanti del linguaggio, dall'uso delle variabili e della segmentazione delle stringhe, ai vari tipi di PRINT per la formattazione dello schermo, alle matrici ecc. L'unica barriera nella produzione di giochi d'alta qualità è la propria immaginazione.

Prima di mettersi a programmare un'avventura, occorre però avere idee ben chiare su cosa si vuol fare e sul soggetto. Trama, ostacoli, trappole e così via devono essere tutti predefiniti per risparmiarsi possibili problemi.

Prima di tutto, buttiamo giù le idee su un foglio: anche se non si ha un quadrao completo di tutte le implicazioni, basta un'idea generale, un'ambientazione e qualche ostacolo sul cammino del giocatore. Nelle prossime lezioni verrà spiegato come tramutare l'idea di un gioco in un programma, imparando a inserire le proprie idee originali. Si faccia molta attenzione al mondo che si sceglie per ambientare l'avventura e si cerchi di renderlo il più interessante possibile: ci vuole un bel po' di fatica per scrivere un'avventura divertente ambientata in un grigio ambiente d'ufficio!

Ci si può ispirare a libri, romanzi, film, commedie, oppure a idee totalmente nuove. Si può prendere a prestito qualcosa da altre storie, anche se la fonte migliore è... un buon incubo! Comunque sia, occorre sempre un tema centrale da sviluppare nel corso dell'avventura.

Si cerchi sempre un giusto equilibrio tra sfida e impossibilità: non val la pena dedicare un sacco di tempo a scrivere un'avventura se poi chiunque riesce a risolverla nel giro di mezz'ora. Viceversa, non ci faremo certo degli amici tra i giocatori, se le nostre storie sono del tutto impossibili da risolvere.

La regola è *dare a chi punta una possibilità, ma non troppe!*

Si cerchi anche di non lasciare troppi ambienti vuoti, che nulla aggiungono all'avventura, mentre occupano spazio vitale in memoria (per non parlare del fastidio che danno!).

Non si parta con storie troppo complicate perché i problemi da affrontare sono complessi finché non si è acquisita buona pratica.

Si valutino bene le conseguenze di ogni tentativo ambizioso e si tenga d'occhio anche la quantità di memoria che rimane alla macchina.

Nell'avventura che viene sviluppata nel corso di Giochi al Computer, troviamo molte frasi REM. Per risparmiare memoria, in un'avventura lunga conviene ometterle del tutto, anche se ciò comporta una più difficile comprensione del listato.

Altro spazio su cui economizzare memoria è quello delle descrizioni degli ambienti (senza peraltro accorciarle troppo, altrimenti l'avventura perde di fascino). Nella prossima lezione vedremo come passare dall'idea dell'avventura a una mappa e impostare il programma.

Quanta memoria serve per scrivere un gioco d'avventura?

La risposta giusta sarebbe: "tutta quella possibile"! L'avventura sviluppata nel corso di questa sezione è molto semplice e occupa circa 5K di BASIC, ma si noterà che non è a un livello confrontabile con quanto reperibile in commercio. Le avventure migliori hanno un gran numero di stanze e di indovinelli e, inoltre, risultano difficili per l'accumularsi dei problemi. Con una quantità di memoria limitata si è costretti a selezionare un numero di problemi minori, ma di maggior complessità, per impedire una soluzione rapida e di scarsa soddisfazione. Un computer con 16K di memoria può considerarsi il minimo.

QUANTA MEMORIA RESTA?

Nello scrivere un lungo gioco d'avventura è facile scoprire che si stanno forzando i limiti di memoria della macchina. È ovvio che il problema è più acuto quanto minore è la memoria della macchina e, del resto, è una perdita di tempo scrivere un avventura per un computer con meno di 16K.

Per ogni macchina esiste un modo di controllare quanta memoria resta. Basta una semplice routine:

Sullo Spectrum, si scriva:

```
Print (PEEK 23730 + 256*PEEK 23731)—
   (PEEK 23653 + 256*PEEK 23654)
```

Così si mette in conto sia il programma in sé che lo spazio delle variabili. Una migliore valutazione della memoria restante si ha appena dopo aver eseguito un RUN del programma (se ciò è possibile nel corso della stesura).

Sui Commodore, si ottiene la memoria restante digitando:

```
PRINT FRE (0)
```

La memoria restante tiene conto sia dello spazio occupato dal programma che dalle variabili, ma in tal caso occorre usare la linea appena descritta dopo aver dato un RUN del programma, se ciò è possibile nel corso della stesura.

Sugli Acorn, si ottiene la memoria restante digitando:

```
PRINT HIMEM — TOP
```

Questo numero comprende solo lo spazio occupato dal programma: lo spazio delle variabili non è valutabile, e di ciò va tenuto conto lasciando spazio sufficiente. Per ottenere l'ammontare massimo di memoria si usi il Mode 7 sul BBC e il Mode 6 sull'Electron.

Sul Dragon e sul Tandy, si trova la memoria restante digitando:

```
PRINT MEM
```

Il risultato tiene conto dello spazio occupato dal programma e dalle variabili, perciò conviene eseguire un RUN nel corso della stesura, se ciò è possibile, per avere un'indicazione reale sulla memoria totale impiegata.

Esiste un metodo molto facile per aumentare la quantità di memoria disponibile sul Dragon per i programmi in BASIC. Prima di cominciare a programmare, si digitino queste due POKE:

```
POKE 25,6
POKE 26,1
NEW
```

Si guadagnano così altri 6Kbyte per il programma, facendo utilizzare al programma BASIC lo spazio normalmente riservato alla grafica ad alta risoluzione, ossia quattro pagine di 1 Kbyte e mezzo.

Se si usano le POKE, ci si accerti di non usare comandi grafici nell'avventura, altrimenti si sciupa il programma. Quando, dopo aver memorizzato il programma su nastro o su disco, lo si vuole ricaricare in memoria, non ci si dimentichi di digitare prima le POKE, seguite da un NEW.

AMBIENTI PER UN'AVVENTURA

La prima tappa, nel progetto di un'avventura, è la stesura della trama di massima e la preparazione di una cartina per ciascun ambiente

Prima di redigere il programma, occorre aver ben impostato l'intera storia, altrimenti è facile commettere errori e ritrovarsi con uno svolgimento slegato e frammentario.

Le prossime sezioni di Giochi al Computer porgono un valido esempio su come sviluppare un tipico (anche se necessariamente breve) programma di avventura. La traccia, per l'avventura qui presentata, consiste nell'affannoso recupero del preziosissimo occhio purpureo di un'icona. Seguendo con attenzione tutti i successivi stadi dello sviluppo, si imparerà rapidamente come scrivere un'avventura per conto proprio.

LA TRAMA

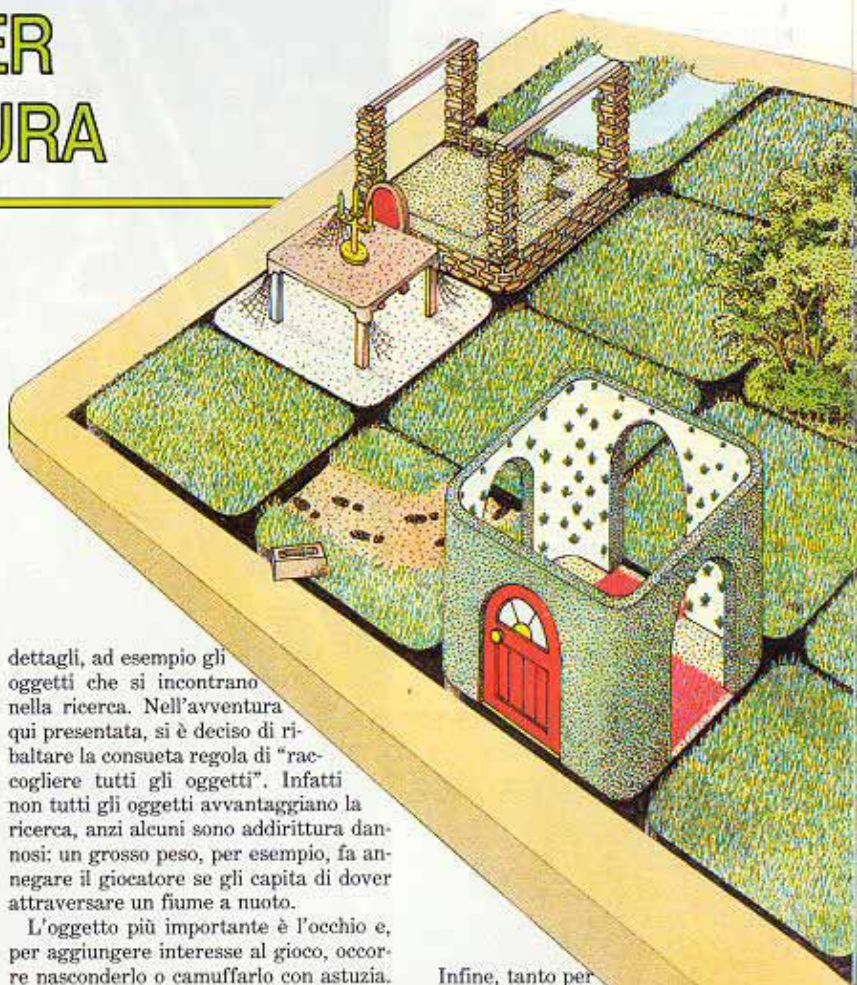
Per progettare l'ambiente più adatto alla traccia della storia, la nostra fantasia deve creare oggetti, luoghi e rompicapo in quantità, tutti con uno scopo ben preciso.

Non occorre fare tutto in una sola mandata, poiché, una volta impostata, la storia si arricchisce man mano di nuove idee, mentre i dettagli si incastrano al posto giusto. All'inizio basta una traccia generale.

In questa avventura, il giocatore, che si trova in una situazione finanziaria "nera", è partito alla ricerca di un favoloso e inestimabile occhio di un'icona, nascosto in qualche parte nel mondo. Per sua sfortuna, il protagonista è inseguito da un'ispettore delle tasse che ha un ruolo simile a quello del pirata in altre avventure, ossia complicare la vita al povero avventuriero. Quando compare l'ispettore delle tasse, possono succedere due cose. Se si ha con sé un oggetto, egli se ne appropria, come rimborso dell'enorme debito con il fisco. Se invece non si ha ancora niente (e non si può pagare) il giocatore finisce in *gattabuia*.

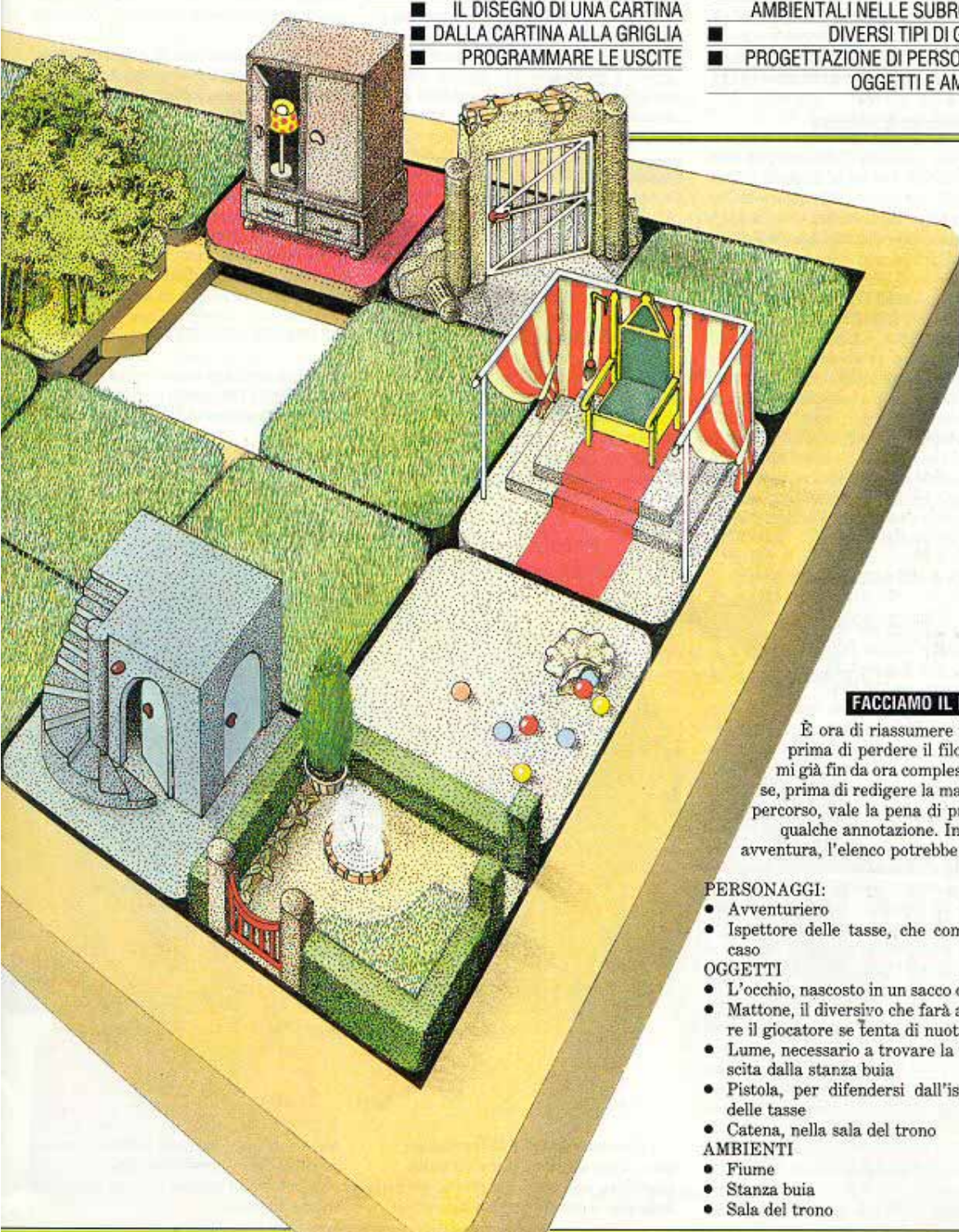
Come *canovaccio*, questa prima traccia può bastare; adesso vanno inseriti alcuni dettagli, ad esempio gli oggetti che si incontrano nella ricerca. Nell'avventura qui presentata, si è deciso di ribaltare la consueta regola di "raccogliere tutti gli oggetti". Infatti non tutti gli oggetti avvantaggiano la ricerca, anzi alcuni sono addirittura dannosi: un grosso peso, per esempio, fa annegare il giocatore se gli capita di dover attraversare un fiume a nuoto.

L'oggetto più importante è l'occhio e, per aggiungere interesse al gioco, occorre nasconderlo o camuffarlo con astuzia. Potremmo nasconderlo in uno scrigno o in una nicchia, ma conviene scegliere sistemi più subdoli, per confondere l'avventuriero: anziché nascondere la pietra nel posto in cui sarebbe logico trovare qualcosa di prezioso, è stata celata in un sacco di palline. L'avventuriero penserà che giocare alle biglie non lo condurrà lontano...

Un ambiente molto comune nei giochi di avventura è la stanza buia in cui possono accadere le cose più tremende: questa avventura non fa eccezione. Senza una torcia, che si incontra altrove nel percorso, il povero giocatore non riesce più a orientarsi e rimane bloccato. Ciò può forse sembrare un po' sleale, non avendo dato alcun avvertimento della triste sorte in un simile frangente, ma altrimenti che avventura sarebbe? Per dare al giocatore una possibilità in più di difendersi, da qualche parte è nascosta un'arma, una pistola o magari un coltello.

Infine, tanto per aggiungere qualcosa di scherzoso, c'è una sala del trono e una catena. La sala, però, può non esser quel che sembra: se non si ha con sé l'occhio, infatti, la sala del trono diventa... una toilette! Tirando la catena, si è sciacquati via fuori dall'avventura!

Resta da mettere a punto un'altra cosa importante: le condizioni per vincere il gioco. Non esistono uscite palesi dall'avventura e uno degli enigmi è proprio come uscire una volta trovata la pietra.

Ovviamente, per la vittoria non basta trovare le biglie: occorre che il giocatore riconosca l'occhio e, per complicare le cose, egli deve trovarsi nella sala del trono e tirare la catena (in tal caso non ha più conseguenze spiacevoli). Il vantaggio di creare un'uscita, per così dire, "bivalente" è che può scoraggiare i prudenti dal tentarla troppe volte.

FACCIAMO IL PUNTO

È ora di riassumere il tutto, prima di perdere il filo dei temi già fin da ora complessi. Forse, prima di redigere la mappa del percorso, vale la pena di prendere qualche annotazione. In questa avventura, l'elenco potrebbe essere:

PERSONAGGI:
- Avventuriero
- Ispettore delle tasse, che compare a caso

OGGETTI
- L'occhio, nascosto in un sacco di biglie
- Mattone, il diversivo che farà annegare il giocatore se tenta di nuotare
- Lume, necessario a trovare la via d'uscita dalla stanza buia
- Pistola, per difendersi dall'ispettore delle tasse
- Catena, nella sala del trono

AMBIENTI
- Fiume
- Stanza buia
- Sala del trono

Fin qui, nell'avventura esistono soltanto tre ambienti in rapporto alle cose che devono trovarvi posto. Già fin da ora si potrebbero inserire altri elementi, ma conviene prima collegare assieme quelli già esistenti, redigendo una mappa dell'avventura.

DISEGNO DELLA CARTINA

Una prima versione della cartina sarà semplicemente una serie di quadrati collegati da frecce, come nella **figura 1**. Ciascun quadrato rappresenta una stanza o un ambiente (un termine quanto mai appropriato, poiché non si è mai confinati al chiuso in questi giochi d'avventura). Un ambiente può essere di qualsiasi tipo: dalla capocchia di uno spillo appuntato sulla veste della regina, a una pianura sterminata. All'elenco preliminare vanno aggiunti tutti gli ambienti iniziali, più altri ancora che servono a collegare le varie fasi del gioco.

Nel disegnare questa cartina ci si rammenti di annotare le direzioni che si possono prendere da ogni stanza, poiché può essere preferibile avere uscite che funzionino a *senso unico*, accompagnate da un messaggio del tipo:

LA PORTA SI RICHIUDE ALLE TUE SPALLE

Le linee tratteggiate che escono dalla stanza buia indicano che il giocatore può seguire quella direzione soltanto se alcune condizioni sono soddisfatte. Nel nostro caso, la condizione è che l'avventuriero abbia il lume con sé e che questo sia acceso.

È difficile prevedere quanti ambienti si possono far entrare in un quantitativo limitato di RAM. La difficoltà deriva dal fatto che molti sono i fattori che concorrono a consumare spazio in un programma d'avventura: descrizioni di ambienti, l'elenco dei comandi, l'uso cui questi son destinati, i rompicapo e via dicendo.

Dopo aver scritto un certo numero di avventure, verificando di volta in volta la memoria occupata (vedere a pagina 268), possiamo farci un'idea di quanto entra nella macchina. Chi ha un computer da 16K vedrà ben presto che, con così poca RAM, è quasi impossibile scrivere un'avventura in grande stile. Comunque, l'avventura sviluppata nel corso delle prossime lezioni ha solo poche stanze, 12 in tutto e perciò non dovrebbe creare problemi.

Una mappa per "La ricerca dell'Occhio di Diamante" è alla **figura 1**. La descrizione degli ambienti, a questo livello, deve mantenersi breve. Sono stati tracciati i collegamenti e indicato il punto di partenza. Quest'ultimo è importante perché riguarda il modo di affrontare l'avventura, l'ordine col quale vengono incontrati gli oggetti e gli enigmi. Gli oggetti sono segnalati nei loro ambienti, mentre quelli che compaiono solo in seguito, come l'occhio, sono elencati di fianco alla mappa.

DALLA CARTINA ALLA GRIGLIA

Completata la mappa, si possono trasferire le informazioni su un reticolo o una griglia. Due sono i tipi di griglie comunemente usati nei giochi di avventura: uno è basato su quadrati come nella **figura 2** e uno su ottagoni come nella **figura 3**. Il tipo da usare dipende dal numero di uscite possibili da ciascun ambiente.

Le avventure più semplici prevedono uscite nelle quattro direzioni Nord, Sud, Est e Ovest (come nell'avventura dell'Occhio). Se un'avventura offre questo arco di possibilità è più comodo adoperare una griglia di quadrati: come procedere, è spiegato in dettaglio più avanti. Se, invece, le uscite comprendono anche Nord Est, Nord Ovest, Sud Est e Sud Ovest, va usata una griglia di ottagoni, molto più complessa.

Un'altra variante per la griglia è nel caso che sia possibile uscire anche verso l'alto e verso il basso: allora, conviene adottare una griglia a parte per ogni livello dell'avventura.

Nell'avventura qui sviluppata viene adoperata una griglia di quadrati (uscite a Nord, Sud, Est e Ovest). Per la maggior parte delle avventure, quattro uscite sono più che sufficienti e si può sempre simulare un movimento verso alto o verso il basso, adottando una descrizione di questo genere:

C'È UNA SCALA CHE SCENDE A OVEST

Questa avventura richiede una griglia di sei quadrati per quattro, cioè il numero massimo di ambienti sulla cartina, dall'al-

1. La prima cartina dell'avventura mostra gli ambienti previsti nelle rispettive posizioni. Le frecce normali indicano le uscite sempre accessibili, mentre quelle a strisce indicano uscite possibili solo a certe condizioni particolari (in questo caso se si ha una torcia accesa)

to in basso e da destra verso sinistra. Prima di trasferire il tutto sulla griglia, si controlli di aver numerato ciascun quadrato partendo dall'alto a sinistra e procedendo fino a destra in basso.

Dopo aver numerato i quadrati e avervi riportato gli ambienti, la griglia appare come alla **figura 4** (pagina 301).

INIZIO DEL PROGRAMMA

Tracciata in linea di massima la trama e completata la griglia, ecco il programma.

Il primo passo è la trascrizione delle descrizioni per gli ambienti, basate sulla griglia. Il testo va scelto accuratamente per creare la giusta atmosfera, ma con il minimo dispendio di memoria. Insieme alle descrizioni degli ambienti, occorre anche specificare le direzioni delle uscite.

Ecco, infine, i primi moduli del programma. La numerazione delle linee è alta (da 5000 in sù) per lasciare spazio sufficiente ad altre sezioni di programma, che verranno aggiunte in seguito.

Trascritta la sezione per il proprio computer, la si conservi sul nastro.

```
5000 REM **DESCRIZIONI AMBIENTI**
5010 REM **AMBIENTE 4**
5020 PRINT "SEI ALL'ESTERNO DI UN
   GRANDE EDIFICIO"
5030 LET N=0: LET E=0: LET S=1: LET
   W=0: RETURN
5040 REM **AMBIENTE 7**
5050 PRINT "SEI ACCANTO AD UN FIUME
   IMPETUOSO"
5060 LET N=0: LET E=1: LET S=0: LET
   W=0: RETURN
5070 REM **AMBIENTE 8**
5080 PRINT "SEI IN UNA FORESTA
   PIETRIFICATA"
5090 LET N=0: LET E=0: LET S=1: LET
   W=1: RETURN
5100 REM **AMBIENTE 10**
5110 PRINT "SEI IN UNA STANZA
   POLVEROSA"
5120 LET N=1: LET E=1: LET S=1: LET
   W=0: RETURN
5130 REM **AMBIENTE 11**
5140 PRINT "SEI IN UNA STANZA BUIA"
5150 IF LA<>1 THEN LET N=0: LET E
   =0: LET S=0: LET W=0: PRINT "È
   TROPPO BUIO PER VEDERE UN'USCITA":
   LET DA=1: RETURN
5160 LET N=0: LET E=0: LET S=1: LET
   W=1: RETURN
5170 REM **AMBIENTE 14**
5180 PRINT "SEI SU SENTIERO FANGOSO"
5190 LET N=1: LET E=1: LET S=0: LET
   W=0: RETURN
5200 REM **AMBIENTE 15**
5210 PRINT "SEI ACCANTO AL
   CANCELLO□□□□□□□DELLA CITTÀ
   SEGRETA"
5220 LET N=0: LET E=1: LET S=0: LET
   W=1: RETURN
5230 REM **AMBIENTE 16**
5240 PRINT "SEI NELL'INGRESSO
   PRINCIPALE"
5250 LET N=1: LET E=1: LET S=1: LET
   W=1: RETURN
5260 REM **AMBIENTE 17**
5270 PRINT "SEI NEL CORTILE"
5280 LET N=1: LET E=1: LET S=0: LET
   W=1: RETURN
5290 REM **AMBIENTE 18**
5300 PRINT "SEI NEL GIARDINO"
5310 LET N=0: LET E=0: LET S=1: LET
   W=1: RETURN
5320 REM **AMBIENTE 22**
```

STANZA BUIA

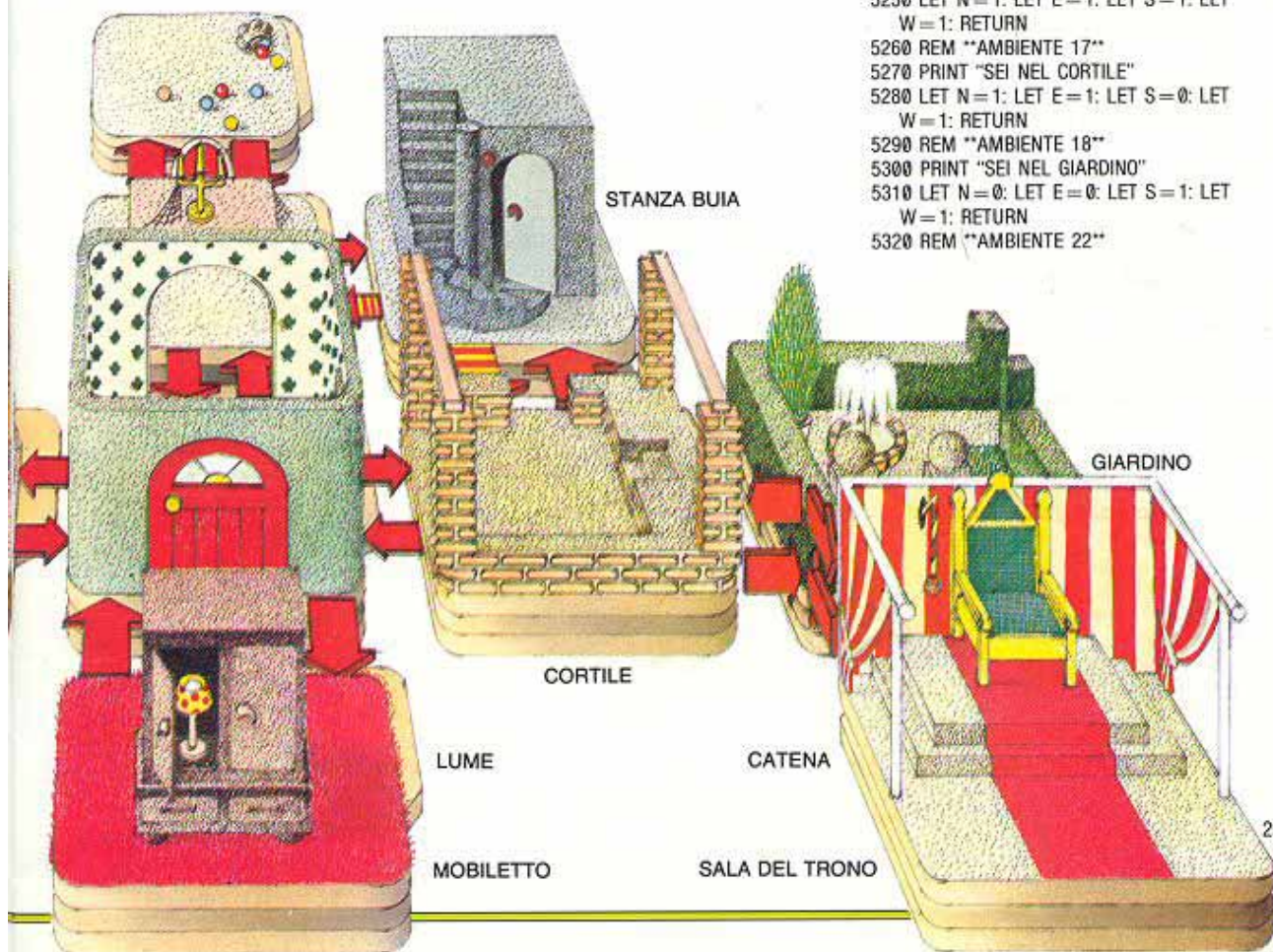
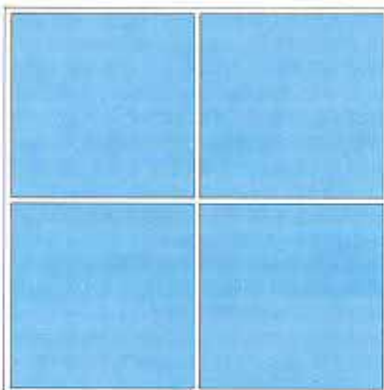
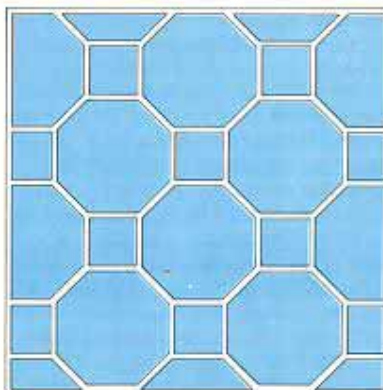
GIARDINO

CORTILE

LUME

CATENA

MOBILETTO

SALA DEL TRONO

2. Un particolare di una griglia di quadrati su cui progettare avventure con quattro uscite, verso Nord, Sud, Est e Ovest rispettivamente

3. Un particolare di una griglia di ottagoni, che si usa quando si vogliono uscite anche verso Nord Ovest, Nord Est, Sud Ovest e Sud Est

```
5330 PRINT "SEI NEL MOBILETTO"
5340 LET N=1: LET E=0: LET S=0: LET
   W=0: RETURN
5350 REM **AMBIENTE 24**
5360 PRINT "SEI NELLA SALA DEL TRONO"
5370 LET N=1: LET E=0: LET S=0: LET
   W=0: RETURN
```

Chi ha il Vic 20 dovrà aggiustare la spaziatura nelle PRINT, per adattare le scritte allo schermo con 22 colonne.

```
5000 REM **DESCRIZIONI AMBIENTI**
5010 REM **AMBIENTE 4**
5020 PRINT "SEI ALL'ESTERNO DI UN
   GRANDE EDIFICIO"
5030 N=0:E=0:S=1:W=0:RETURN
5040 REM **AMBIENTE 7**
5050 PRINT "SEI ACCANTO AD UN FIUME
   IMPETUOSO"
5060 N=0:E=1:S=0:W=0:RETURN
5070 REM ** AMBIENTE 8 **
5080 PRINT "SEI IN UNA FORESTA
   PIETRIFICATA"
5090 N=0:E=0:S=1:W=1:RETURN
5100 REM ** AMBIENTE 10 **
5110 PRINT "SEI IN UNA STANZA
   POLVEROSA"
5120 N=1:E=1:S=1:W=0:RETURN
5130 REM ** AMBIENTE 11 **
5140 PRINT "SEI IN UNA STANZA BUIA"
5150 IF OB(6)<>-1 OR LA<>
   1 THEN N=0:E=0:S=0:W=0
5155 IF OB(6)<>-1 OR LA<>1 THEN
   PRINT "È TROPPO BUIO PER VEDERE
   UN'USCITA":RETURN
5160 N=0:E=0:S=1:W=1:RETURN
5170 REM ** AMBIENTE 14 **
5180 PRINT "SEI SU SENTIERO FANGOSO"
5190 N=1:E=1:S=0:W=0:RETURN
5200 REM ** AMBIENTE 15 **
5210 PRINT "SEI ACCANTO AL CANCELLO
   DELLA CITTÀ SEGRETA"
5220 N=0:E=1:S=0:W=1:RETURN
5230 REM ** AMBIENTE 16 **
5240 PRINT "SEI NELL'INGRESSO
   PRINCIPALE"
5250 N=1:E=1:S=1:W=1:RETURN
5260 REM ** AMBIENTE 17 **
5270 PRINT "SEI NEL CORTILE"
5280 N=1:E=1:S=0:W=1:RETURN
5290 REM ** AMBIENTE 18 **
5300 PRINT "SEI NEL GIARDINO"
5310 N=0:E=0:S=1:W=1:RETURN
5320 REM ** AMBIENTE 22 **
5330 PRINT "SEI NEL MOBILETTO"
5340 N=1:E=0:S=0:W=0:RETURN
5350 REM ** AMBIENTE 24 **
5360 PRINT "SEI NELLA SALA DEL TRONO"
5370 N=1:E=0:S=0:W=0:RETURN
```

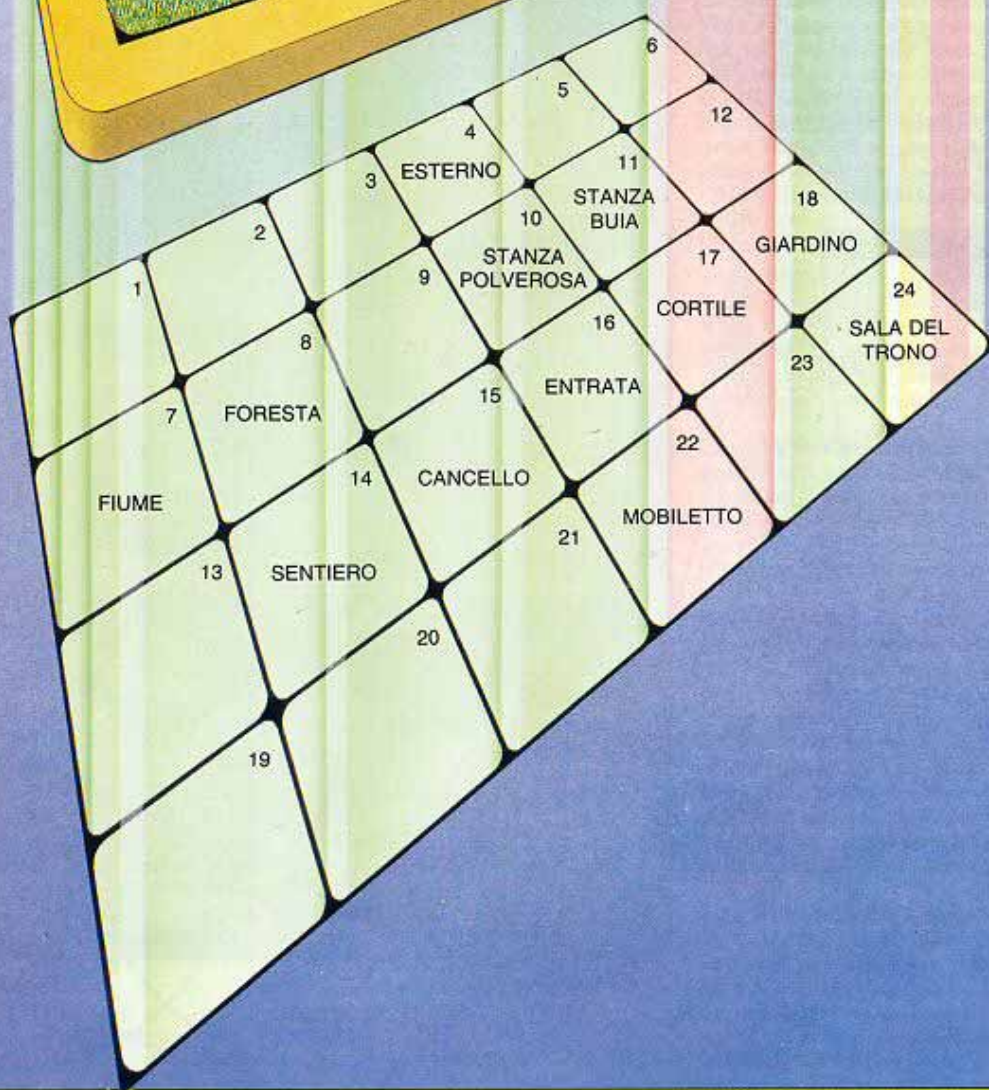
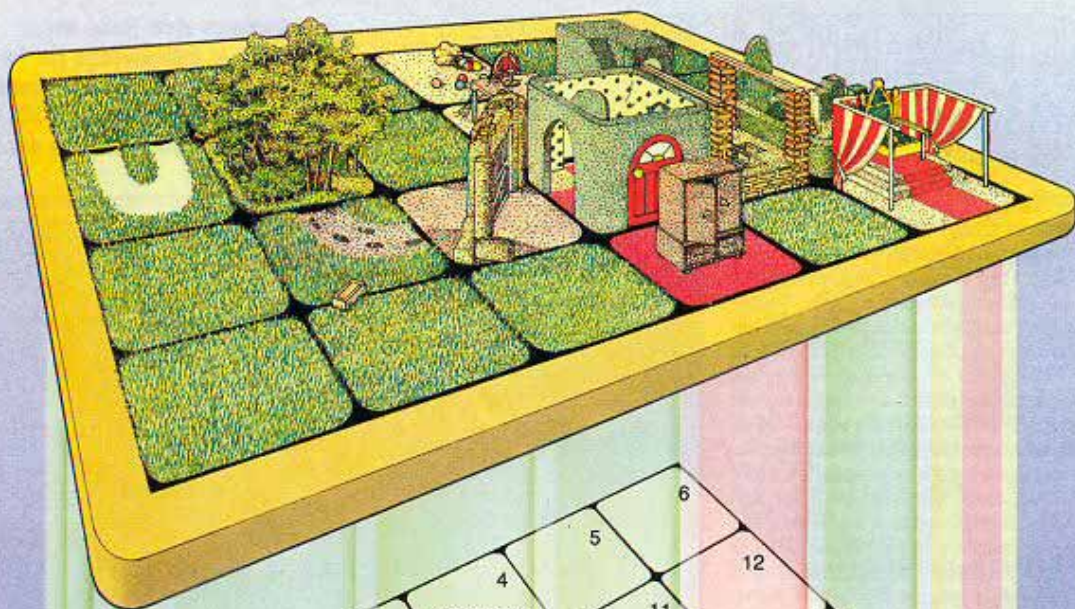
Per gli apparecchi Acorn, si omettano gli spazi alla linea 5210.

```
5000 REM**DESCRIZIONI AMBIENTI**
5010 REM ** AMBIENTE 4**
5020 PRINT "SEI ALL'ESTERNO DI UN
   GRANDE EDIFICIO"
5030 N=0:E=0:S=1:W=0:RETURN
5040 REM **AMBIENTE 7**
5050 PRINT "SEI ACCANTO AD UN FIUME
   IMPETUOSO"
5060 N=0:E=1:S=0:W=0:RETURN
5070 REM **AMBIENTE 8**
5080 PRINT "SEI IN UNA FORESTA
   PIETRIFICATA"
5090 N=0:E=0:S=1:W=1:RETURN
5100 REM **AMBIENTE 10**
5110 PRINT "SEI IN UNA STANZA
   POLVEROSA"
5120 N=1:E=1:S=1:W=0:RETURN
5130 REM **AMBIENTE 11**
5140 PRINT "SEI IN UNA STANZA BUIA"
5150 IF OB(6)<>-1 OR LA<>1 THEN
   N=0:E=0:S=0:W=0:PRINT"È TROPPO
   BUIO PER VEDERE UN'USCITA": RETURN
5160 N=0:E=0:S=1:W=1:RETURN
5170 REM **AMBIENTE 14**
5180 PRINT "SEI SU SENTIERO FANGOSO"
5190 N=1:E=1:S=0:W=0:RETURN
5200 REM **AMBIENTE 15**
5210 PRINT "SEI ACCANTO AL
   CANCELLO□□□□□□DELLA CITTÀ
   SEGRETA"
5220 N=0:E=1:S=0:W=1:RETURN
5230 REM**AMBIENTE 16**
5240 PRINT "SEI NELL'INGRESSO
   PRINCIPALE"
5250 N=1:E=1:S=1:W=1:RETURN
5260 REM**AMBIENTE 17**
5270 PRINT "SEI NEL CORTILE"
5280 N=1:E=1:S=0:W=1:RETURN
5290 REM**AMBIENTE 18**
5300 PRINT «SEI NEL GIARDINO"
5310 N=0:E=0:S=1:W=1:RETURN
5320 REM **AMBIENTE 22**
5330 PRINT "SEI NEL MOBILETTO"
5340 N=1:E=0:S=0:W=0:RETURN
5350 REM **AMBIENTE 24**
5360 PRINT "SEI NELLA SALA DEL TRONO"
5370 N=1:E=0:S=0:W=0:RETURN
```

Non spaventi la gran profusione di frasi REM, voracissime di memoria: a questo livello sono indispensabili per illustrare le varie parti del programma oppure a quale numero d'ambiente si riferisca una particolare descrizione. Più avanti si possono eliminare. A ogni linea di descrizione segue una linea che riguarda le possibili uscite dall'ambiente. Le variabili N, S, E e O (Nord, Sud, Est e Ovest) possono assumere soltanto uno tra due valori: 0 significa che non c'è uscita in quella direzione, 1 che l'uscita c'è.

Infine, si notino le RETURN, al termine di ogni sezione del programma: ogni descrizione d'ambiente è richiamata da una GOSUB. La IF ... THEN, nella sezione per la stanza buia, verifica se il giocatore ha il lume, ma una descrizione delle variabili usate viene data più avanti, assieme a quella degli oggetti, mentre la prossima lezione riguarda il movimento dell'avventuriero attraverso gli ambienti.

4. Lo stadio finale prima di iniziare a programmare: la griglia ricalca la cartina, ma in forma più funzionale

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
ESTERNO
STANZA BUIA
STANZA POLVEROSA
GIARDINO
CORTILE
SALA DEL TRONO
ENTRATA
FORESTA
CANCELLO
FIUME
MOBILETTO
SENTIERO

AVVENTURE MOVIMENTATE

Nei giochi d'avventura, è divertente esplorare nuovi e strani mondi, senza muoversi da casa. Vediamo come partire in queste esplorazioni

Dopo aver trascritto il programma con tutte le descrizioni degli ambienti, permettiamo ora al giocatore di esplorarli, spostandosi da un luogo all'altro. In ogni fase occorre fissare quali siano i movimenti possibili e porre dei limiti, valutando difficoltà e quantità di indizi da raccogliere durante l'avventura.

Stavolta, nell'espansione del programma d'avventura in via di sviluppo, useremo routine che visualizzano la descrizione degli ambienti assieme alle possibili uscite da essi. Al giocatore viene chiesto di impartire una direttiva, cosicché si imparerà a scrivere una sezione di programma per spostare il giocatore all'interno del fantastico mondo, in base alla scelta fatta.

LA PARTENZA

In ogni momento del gioco, il computer deve sempre sapere dove si trovi il cacciatore di avventure e questa informazione è contenuta nella variabile L (per luogo). Alla variabile viene assegnato un valore pari a quello dell'ambiente in cui si sposta a ogni mossa. Per cominciare l'avventura, si tratta di comunicare al computer l'ambiente di partenza del giocatore.

Qui di seguito c'è la sezione di programma che ha tale scopo. Si legga da nastro la sezione di programma trascritta nella lezione precedente, e si aggiungano queste linee:

```
100 CLS: LET DA=0: LET TA=0: LET LA
    =0
270 REM **POSIZIONE DI PARTENZA**
280 LET L=15
290 GOTO 330
```

```
270 REM ** POSIZIONE DI PARTENZA **
280 L=15
290 GOTO 330
```

```
10 TA=0:LA=0
100 MODE 6
270 REM**POSIZIONE DI PARTENZA**
280 L=15
290 GOTO 330
```

Il numero 15 corrisponde all'ambiente in cui si trova il cancello per la città nascosta. Volendo far partire l'avventura da un luogo diverso, basta modificare il valore di L. Tra breve si vedrà come variare il valore di L a seconda dell'ambiente attraversato.

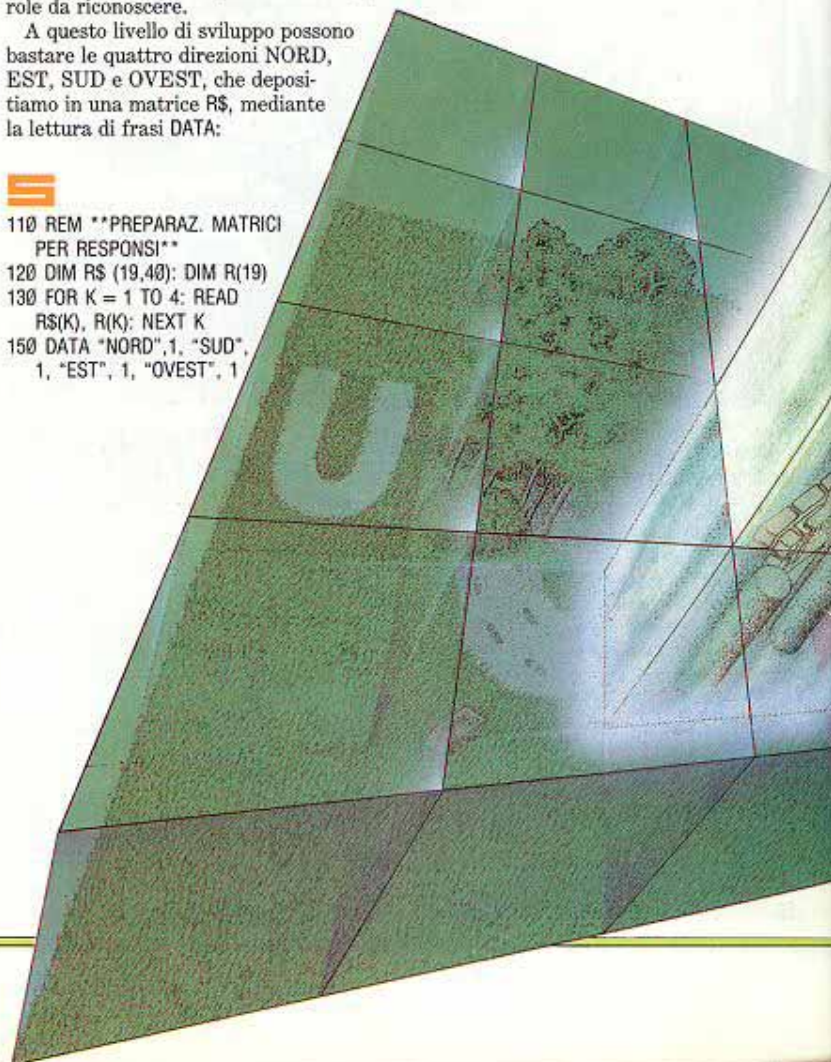
Prima di muovere il cacciatore di avventure, però, occorre comunicare al computer la direzione da intraprendere.

VERIFICA DELLE RISPOSTE

Affinché il computer possa comprendere le direttive del giocatore e agire di conseguenza, occorre fornirgli un elenco di parole da riconoscere.

A questo livello di sviluppo possono bastare le quattro direzioni NORD, EST, SUD e OVEST, che depositiamo in una matrice R$, mediante la lettura di frasi DATA:

```
110 REM **PREPARAZ. MATRICI
    PER RESPONSI**
120 DIM R$ (19,40): DIM R(19)
130 FOR K = 1 TO 4: READ
    R$(K), R(K): NEXT K
150 DATA "NORD",1, "SUD",
    1, "EST", 1, "OVEST", 1
```

```
110 REM**PREPARAZ. MATRICI PER
    RESPONSI"
120 DIM R$(19),R(19)
130 FOR K=1 TO 4:READ R$(K), R(K):NEXT
150 DATA NORD, 1, SUD, 1, EST, 1, OVEST,
    1
```

- VISUALIZZARE DESCRIZIONI E DIREZIONI PER GLI AMBIENTI
- TRATTAMENTO DEI COMANDI
- MUOVERSI PER GLI AMBIENTI
- BLOCCARE I MOVIMENTI IMPOSSIBILI

La linea 120 dimensiona opportunamente le matrici, per contenere tutte le risposte necessarie al gioco. Dato che, per ora, non servono altro che le direzioni, si useranno i primi quattro elementi delle matrici R$ e R.

Il ciclo FOR ... NEXT, alla linea 130, va da 1 a 4 e serve per trasferire le direzioni e i valori ad esse associati (vedere le frasi DATA alla linea 150) nelle matrici R$ e R. Ovviamente, queste informazioni servono a ben poco, se prima il computer non comunica al giocatore dove egli si trovi.

TROVARE UN AMBIENTE

Perché il cacciatore d'avventure sappia dove è finito, dopo ogni spostamento, occorre visualizzare una descrizione dell'ambiente. Le descrizioni sono già state trascritte, quindi serve soltanto una routine che estragga la descrizione abbinata al valore di L, il numero dell'ambiente.

Per questa fase, tornano utili le frasi REM scritte la volta precedente.

I possessori di Spectrum devono aggiungere qualche linea in più, oltre alla routine:

```
20 DIM G(11,4): POKE 23658,8
30 FOR N=1 TO 4: FOR M=1 TO 11: READ
   G(M,N): NEXT M: NEXT N
40 DATA 0,0,0,5020,0,0,5050,5080,0,5110,0
50 DATA 5140,0,0,5180,5210,5240,5270,
   5300,0,0,0
60 DATA 0,5330,0,5360,0,0,0,0,0,0,0
70 DATA 1010,1150,1240,1310,1410,1460,
   1500,1360,1080,1550,3110
300 REM **TROVA AMBIENTE**
310 CLS
330 IF L<11 THEN GOSUB G(L,1): GOTO
   400
340 IF L<21 THEN GOSUB G(L-10,2):
   GOTO 400
350 IF L<26 THEN GOSUB G(L-20,3)
```

```
300 REM ** TROVA AMBIENTE**
310 CLS
330 IF L<11 THEN ON L GOSUB 0,0,0,
    5020,0,0,5050,5080,0,5110:GOTO 400
340 IF L<21 THEN ON L-
    10 GOSUB 5140,0,0,5180,5210,5240,
    5270,5300,0,0:GOTO 400
350 IF L<26 THEN ON L-20 GOSUB 0,
    5330,0,5360
```

Sul Commodore e sul Vic si cambi la linea 310 in:

```
310 PRINT"[]"
```

Prima di poter scrivere per intera questa routine, occorre verificare il numero associato a ciascuna descrizione. A partire dall'ambiente 1, si prenda nota dei numeri di linea relativi a ogni descrizione. Se per un particolare ambiente non esiste descrizione, si annoti uno 0. Nella nostra avventura, per esempio, non ci sono descrizioni per gli ambienti 1,2 e 3, ma ne esiste una per il 4. Completato questo elenco, si può passare alla routine, che, su tutti gli apparecchi escluso lo Spectrum, consiste in una semplice sequenza di operazioni di controllo sul valore di L, tramite l'impiego di ON ... GOSUB. Sullo Spectrum, che non ha questo comando, i numeri di linea vanno inseriti in una matrice come nell'esempio riportato sopra.

Nelle versioni per Dragon, Tandy, Acorn e Commodore, l'elenco dei numeri di linea va dalla linea 330 alla 350 (l'ambiente 1 all'inizio della linea 330 e l'ambiente 24 al termine della linea 350).

Lo Spectrum sfrutta il valore di L per scegliere un elemento dalla matrice preparata dalle linee 20 e 30. Si noterà che nella matrice c'è qualche numero in più rispetto agli ambienti. Tutti questi numeri sono zeri e quindi non influenzano l'operazione del programma, anche se sono necessari perché la matrice viene adoperata in altre parti del programma (le aggiunte si vedranno in un'altra lezione).

I possessori di Spectrum sappiano, inoltre, che la POKE alla linea 20 serve solo ad abilitare i caratteri maiuscoli, cosicché le direttive immesse dal giocatore si accordino alle risposte contenute in R$.

VISUALIZZARE LE DIREZIONI

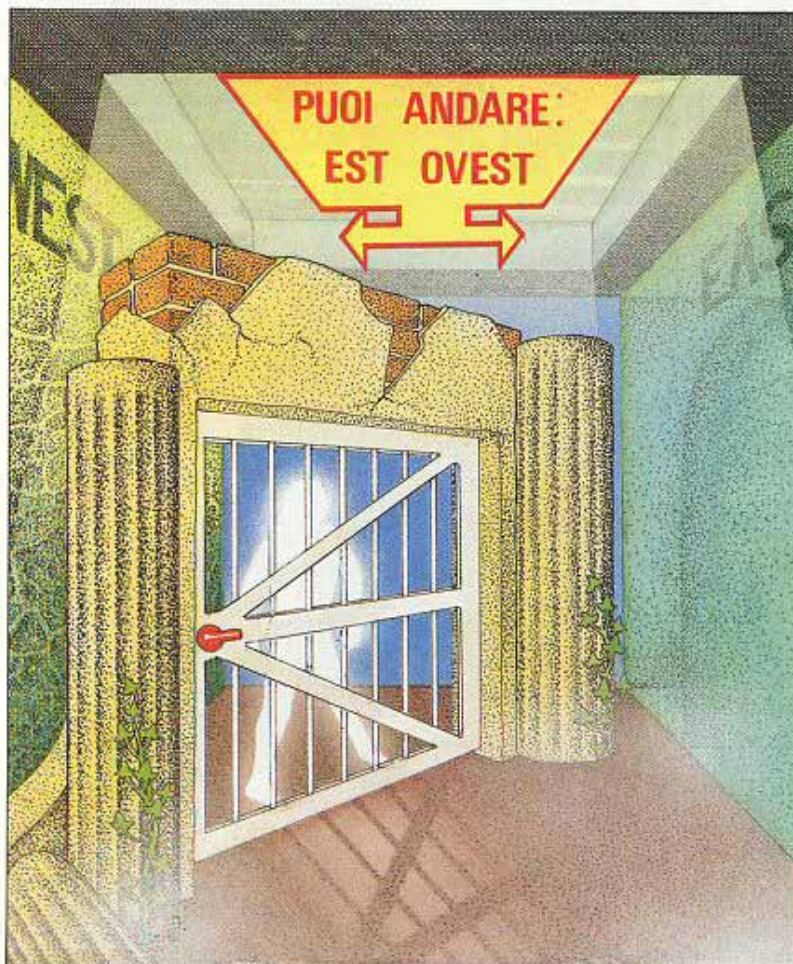
Oltre alle descrizioni degli ambienti, il giocatore vorrà sapere quali uscite ci siano. Siccome non sempre è consentito muoversi in tutte le direzioni, il programma deve aver modo di controllare le informazioni associate alle descrizioni, ossia le variabili N,E,S, W. Le direzioni consentite sono visualizzate da questa routine:

```
390 REM **VISIONE DIREZIONI**
400 IF DA<>1 THEN PRINT'"PUOI
    ANDARE A";
410 IF N>0 THEN PRINT TAB 11; "NORD"
420 IF E>0 THEN PRINT TAB 11; "EST"
430 IF S>0 THEN PRINT TAB 11; "SUD"
440 IF W>0 THEN PRINT TAB 11; "OVEST"
```

```
390 REM **VISIONE DIREZIONI **
400 IF L<>11 OR (LA=1 AND OB(6)=
    -1) THEN PRINT:PRINT "PUOI
    ANDARE A ";:GOTO 410
405 GOTO 460
410 IF N>0 THEN PRINT TAB(11); "NORD"
420 IF E>0 THEN PRINT TAB(11);"EST"
430 IF S>0 THEN PRINT TAB(11);"SUD"
440 IF W>0 THEN PRINT TAB(11);"OVEST"
```

```
390 REM **VISIONE DIREZIONI**
400 IF L<>11 OR (LA=1 AND OB(6)=
    -1) THENPRINT:PRINT "PUOI ANDARE
    A"; ELSE 460
410 IF N>0 THEN PRINT TAB(11);"NORD"
420 IF E>0 THEN PRINT TAB(11);"EST"
430 IF S>0 THEN PRINT TAB(11);"SUD"
440 IF W>0 THEN PRINT TAB(11);"OVEST"
```

Questa sezione verifica il valore di N,S,E e W, basandosi sulla mappa degli ambienti: se il valore delle variabili è maggiore di zero, la direzione è possibile e ciò viene segnalato.

La routine può essere incorporata così com'è in qualsiasi avventura basata su una griglia di quadrati.

LE DIRETTIVE

Ora che il giocatore conosce le direzioni possibili, bisogna dare qualche suggerimento. Questa sezione del programma chiede al giocarore: CHE FAI?

```
450 REM ** RICHIESTA COMANDO**
460 INPUT INVERSE 1;"CHE FAI□"; LINE I$
470 GOSUB 3010
515 GOTO G(I,4)
```

```
450 REM ** RICHIESTA COMANDO **
460 PRINT: INPUT "▀CHE FAIπ"; I$
470 GOSUB 3010
```

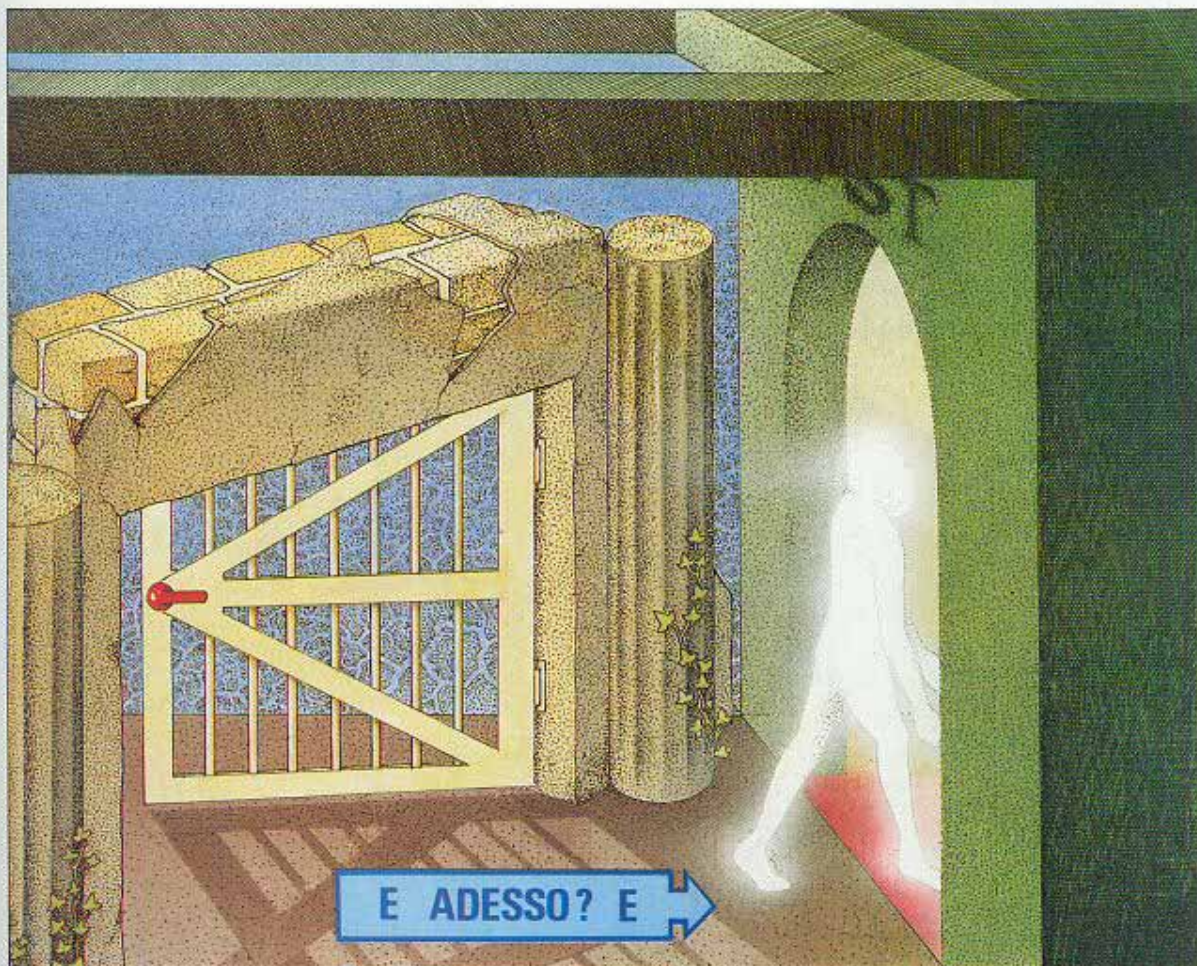
```
450 REM**RICHIESTA COMANDO**
460 INPUT "CHE FAI",I$
470 GOSUB 3010
```

```
450 REM **RICHIESTA COMANDO**
460 PRINT: INPUT "CHE FAI";I$
470 GOSUB 3010
```

Questa è una banale operazione di immissione, in cui la risposta è assegnata a I$. Il computer deve verificarne la correttezza, per cui la linea 470 manda il programma a una subroutine (linea 3010), che elabora la direttiva fornita dal giocatore.

```
600 REM **ROUTINE DI VERIFICA**
610 LET IN=0: IF LEN Y$>LEN X$ THEN
   RETURN
620 FOR K=1 TO (LEN X$-LEN Y$+1)
630 IF Y$=X$(K TO K+LEN Y$-1) THEN
   LET IN=K: GOTO 650
640 NEXT K
650 RETURN
3000 REM **VERIFICA COMANDO**
3010 LET N$="": LET X$=I$: LET Y$
   ="□": GOSUB 600: LET I=IN
3020 IF I=0 THEN LET V$=I$: GOTO 3050
3030 LET V$=I$(TO I-1)
3040 LET N$=I$(I+1 TO)
3050 LET I=0
3060 FOR K=1 TO 19
3070 IF V$=R$(K, TO LEN V$) THEN LET I
   =R(K): LET I$=V$(TO 1)
3080 NEXT K
3090 RETURN
```

```
3000 REM ** VERIFICA COMANDO **
3010 N$="":FOR Z=1 TO LEN(I$):IF MID$
   (I$,Z,1)="□" THEN I=Z:GOTO3020
3015 NEXT:I=0
3020 IF I=0 THEN V$=I$:GOTO 3050
3030 V$=LEFT$(I$,I-1)
3040 N$=MID$(I$,I+1)
```

E ADESSO? E

```
3050 I=0
3060 FOR K=1 TO 19
3070 IF V$=LEFT$(R$(K),LEN(V$)) THEN I
   =R(K):I$=LEFT$(V$,1)
3080 NEXT
3090 RETURN
```

```
3000 REM**VERIFICA DI COMANDO**
3010 N$="":I=INSTR(I$,"□")
3020 IF I=0 THEN V$=I$:GOTO 3050
3030 V$=LEFT$(I$,I-1)
3040 N$=MID$(I$,I+1)
3050 I=0
3060 FOR K=1 TO 19
3070 IF INSTR(R$(K),V$)=1 THEN I=R(K):
   I$=LEFT$(V$,1)
3080 NEXT
3090 RETURN
```

La routine verifica se I$ consiste di due parole: se sì, la prima parola è chiamata V$ e la seconda N$. V$ sta per il verbo, per esempio VADO, UCCIDO o RACCOLGO e per tutte le direzioni come NORD, SUD, EST e OVEST. N$ sta per i nomi degli oggetti nell'avventura.

Dragon, Tandy e Acorn usano INSTR, alla linea 3010, per cercare uno spazio in I$, tra N$ e V$. Il Commodore, il Vic e lo Spectrum non hanno INSTR. Il programma per il Commodore usa MID$ per la ricerca (linee 3010 e 3015), mentre sullo Spectrum la funzione INSTR è sostituita da una piccola subroutine alle linee 600-650.

Se c'è uno spazio, la linea 3030 suddivide le de parti di I$ in N$ e V$. Se non viene trovato uno spazio, la linea 3020 assegna a V$ l'intera I$.

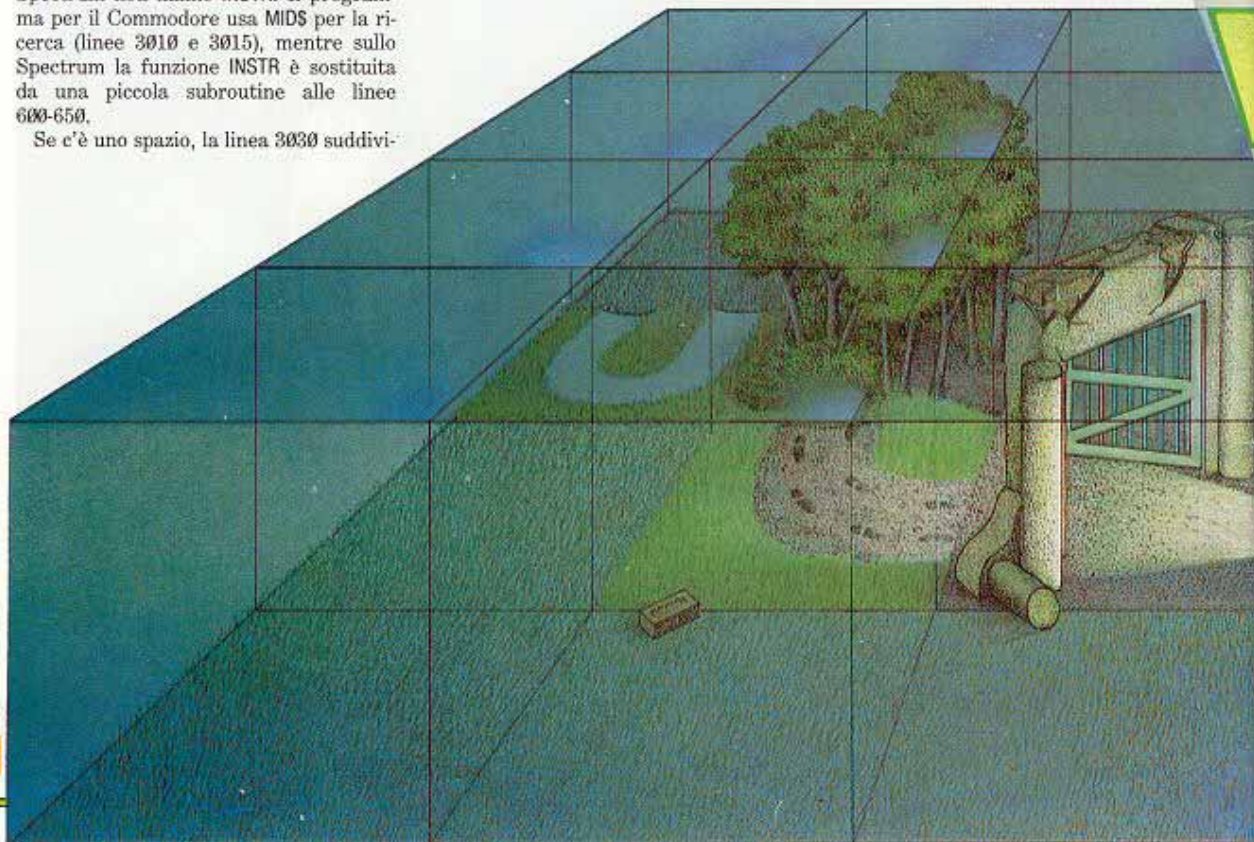
Il resto della subroutine (linee 3060-3080) confronta le direttive con le parole contenute nella matrice R$. Più avanti si vedrà come far sì che essa contenga anche una serie di verbi. Se la linea 3070 trova un abbinamento, a I è assegnato il corrispondente valore di R. In seguito, il computer saprà se l'abbinamento c'è stato, semplicemente verificando se I è maggiore di zero. L'ultima parte della linea 3070 prende la prima lettera di V$, assegnadola a I$, che servirà poi a muovere il cacciatore d'avventure.

Le subroutine possono essere utilizzate, quasi senza modifiche, per ogni gioco d'avventura. L'unica corrrezione, forse, riguarderà la lunghezza del ciclo FOR ... NEXT alla linea 3060.

IL MOVIMENTO

Quello che basta aggiungere per far muover il giocatore in tutti gli ambienti è una routine che trasformi la variabile L in base al contenuto di I$. Eccola:

```
1000 REM **ROUTINE DI MOVIMENTO**
1010 IF I$="N" AND N>0 THEN LET L=L
   -6: GOTO 310
1020 IF I$="E" AND E>0 THEN LET L=L
   +1: GOTO 310
1030 IF I$="S" AND S>0 THEN LET L=L
   +6: GOTO 310
1040 IF I$="O" AND O>0 THEN LET L
   =L-1: GOTO 310
1050 REM **SE DIREZIONE NON È
   POSSIBILE**
1060 PRINT'"SPIACENTE, NON PUOI
   ANDARCI!!": GOTO 330
```

```
1000 REM **ROUTINE DI MOVIMENTO**
1010 IF I$="N" AND N>0 THEN L=L-6:
   GOTO 310
```

```
1020 IF I$="E" AND E>0 THEN L=L+1:
   GOTO 310
1030 IF I$="S" AND S>0 THEN L=L+6:
   GOTO 310
1040 IF I$="O" AND O>0 THEN L=L
   -1: GOTO 310
1050 REM **SE DIREZIONE NON È
   POSSIBILE**
1060 PRINT:PRINT "SPIACENTE, NON PUOI
   ANDARCI!!": GOTO 330
```

```
1000 REM"MOVIMENTO"
1010 IF I$="N" AND N>0 THEN L=L
   -6:GOTO 30
1020 IF I$="E" AND E>0 THEN L=L
   +1:GOTO 310
1030 IF I$="S" AND S>0 THEN L=L+6:
   GOTO 310
1040 IF I$="O" AND O>0 THEN L=L
   -1:GOTO 310
1050 REM****DIREZ. IMPOSSIBILE ****
1060 PRINT "SPIACENTE, NON PUOI
   ANDARCI!!":GOTO 330
```

Si ricorderà che l'avventura è basata su una griglia di sei ambienti. Il movimento implica un cambiamento del valore di L, che dipende dalla grandezza della griglia.

Muoversi a Nord o a Sud comporta aggiungere o sottrarre 6 da L, per salire o scendere di una intera linea sulla griglia. Analogamente, il movimento a Est o a Ovest comporta aggiungere o sottrarre 1 da L.

Le linee da 1010 e 1040 controllano la direzione in I$ e aggiustano L in conseguenza, purché esista un'uscita che si accordi con I$. Le uscite sono state definite nelle linee subito dopo le descrizioni.

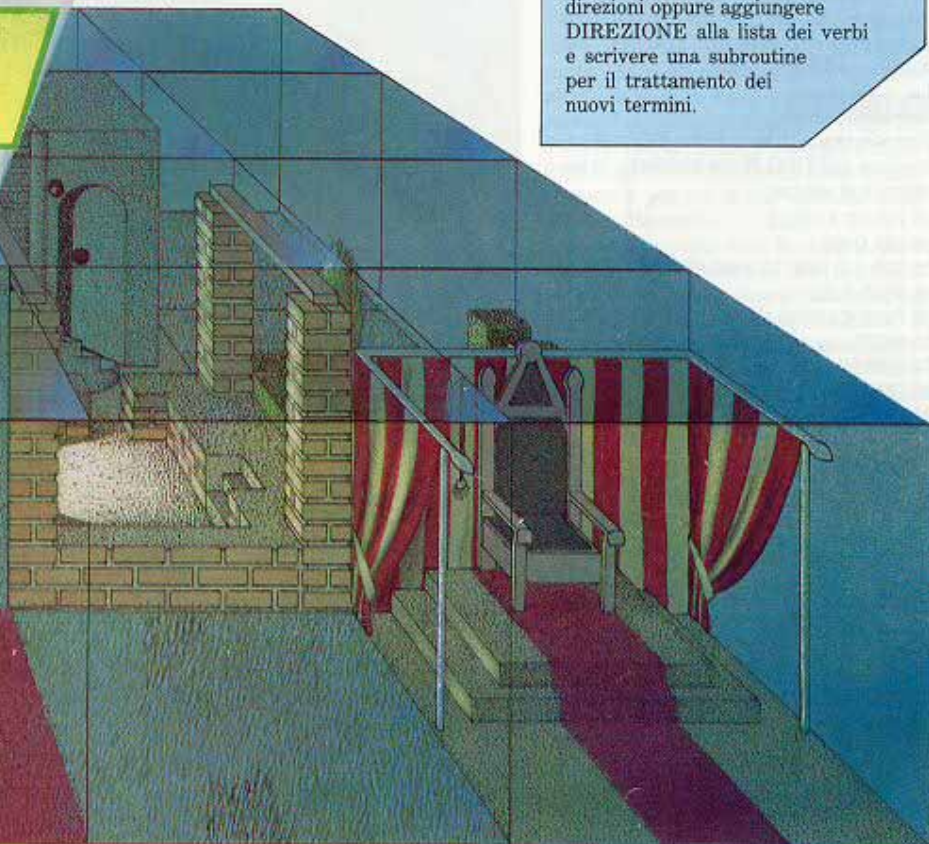
Se non c'è uscita nella direzione scelta, la linea 60 visualizza "SPIACENTE ... NON PUOI ANDARCI!!"

L'unico cambiamento che si dovrà fare, volendo adoperare la routine in un'avventura diversa, riguarda l'eventuale dimensione della griglia. In quel caso andranno modificate le linee 1010 e 1030.

Adesso si memorizzi il programma trascritto, prima di passare alla prossima sezione di Giochi al Computer.

Il programma prevede direttive "NORD", "DIREZIONE NORD", o soltanto le iniziali?

La routine per il controllo delle direttive, dalla linea 3000 alla 3010, è scritta apposta per accogliere parole di una sola lettera, trattandole separatamente dalle direttive più lunghe. Le direttive di una sola lettera sono chiamate I$ e, più oltre, il controllo di I$ avverrà sulle lettere N,S,E e O, permettendo di indicare direzioni "abbreviate". Ciò sveltisce il gioco. Ma nulla impedisce di apportare modifiche, consentendo di usare NORD o DIREZIONE NORD. Più avanti è illustrato come il programma elabori verbi e nomi.
La cosa da fare è aggiungere alla linea le parole complete delle direzioni oppure aggiungere DIREZIONE alla lista dei verbi e scrivere una subroutine per il trattamento dei nuovi termini.

ORA TI TROVI ALL'ENTRATA

UNA MIRIADE DI OGGETTI

Il mondo dell'avventura va riempito di oggetti: ecco come immettere nel programma un elenco con tutte le voci e come sfruttarlo durante il gioco

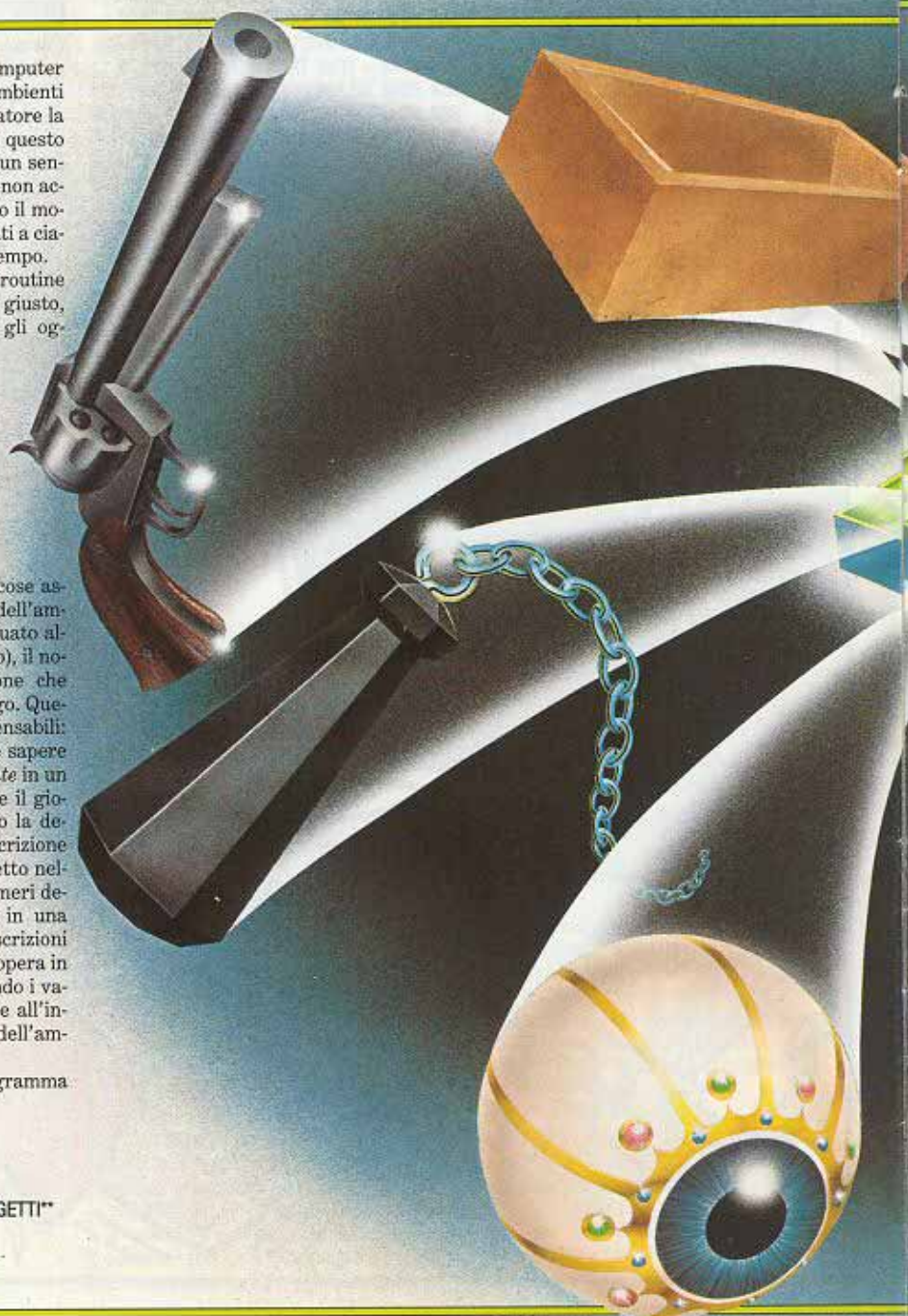
Nell'ultima lezione di Giochi al Computer si è creata una serie completa di ambienti per l'avventura e si è data al giocatore la possibilità di percorrerla tutta. A questo livello tuttavia, ciò non riveste alcun senso, considerato che negli ambienti non accade ancora proprio nulla: è giunto il momento di inserire gli eventi collegati a ciascun ambiente, progettati a suo tempo.

Vediamo come aggiungere le routine necessarie per collocare al posto giusto, raccogliere oppure abbandonare gli oggetti dell'avventura. Viene anche presentata una routine che visualizza un inventario degli oggetti che si hanno con sé, utile nei momenti più critici dell'azione. Carichiamo il programma dell'ultima lezione, per poi trascrivere le nuove routine.

OGGETTI

La macchina deve conoscere tre cose associate agli oggetti: il numero dell'ambiente dove ognuno di essi era situato all'inizio (pronto per essere scoperto), il nome dell'oggetto e una descrizione che suggerisca qualcosa sul suo impiego. Questi tre elementi sono tutti indispensabili: il computer, per prima cosa, deve sapere se un particolare oggetto è *presente* in un ambiente, poi deve poter avvisare il giocatore della sua presenza (usando la descrizione *lunga*). Infine, la descrizione *breve* serve per identificare l'oggetto nelle direttive e nell'inventario. I numeri degli ambienti vengono depositati in una matrice, i nomi in un'altra e le descrizioni in un'altra ancora. Il programma opera in parallelo sulle tre matrici, ricavando i vari elementi semplicemente in base all'indice, che corrisponde al numero dell'ambiente attraversato.

Si aggiungano adesso al programma queste linee:

```
160 REM **PREP. MATRICI DEGLI OGGETTI**
170 READ NB
180 DIM B(NB): DIM B$(NB,14): DIM
    S$(NB,40)
```

- LE FRASI DATA PER GLI OGGETTI
- LA LUNGHEZZA DELLE DESCRIZIONI
- VISUALIZZARE GLI OGGETTI NEGLI AMBIENTI GIUSTI
- AGGIUNGERE I VERBI
- SCELTA DELLA CORRETTA ROUTINE
- PRENDERE E LASCIARE GLI OGGETTI
- AGGIUNGERE UNA ROUTINE CHE FORNISCA L'INVENTARIO

```
190 FOR I=1 TO NB: READ B(I),B$(I),S$(I):
    NEXT I
200 DATA 7,4,"SACCHETTO","C'È UN
    SACCHETTO DI BIGLIE"
210 DATA 14,"MATTONE","C'È UN MATTONE
    AI TUOI PIEDI"
220 DATA 24,"CATENA","UNA CATENA
    PENDE DAL SOFFITTO"
230 DATA 0,"PISTOLA","C'È UNA PISTOLA
    SUL PAVIMENTO"
240 DATA 0,"OCCHIO","PER TERRA C'È UN
    GIOIELLO A □□FORMA DI OCCHIO"
250 DATA 22,"LUME","VEDI UN LUME"
260 DATA 0,"ISPETTORE DELLE TASSE","È
    APPARSO UN ISPETTORE DELLE TASSE!"
```

```
160 REM **PREP. MATRICI DEGLI OGGETTI**
170 READ NB
180 DIM OB(NB),OB$(NB),SI$(NB)
190 FOR I=1 TO NB: READ
    OB(I),OB$(I),SI$(I):NEXT
200 DATA 7,4,SACCHETTO,C'È UN
    SACCHETTO DI BIGLIE
210 DATA 14,MATTONE,C'È UN MATTONE AI
    TUOI PIEDI
220 DATA 24,CATENA, UNA CATENA PENDE
    DAL SOFFITTO
230 DATA 0,PISTOLA,C'È UNA PISTOLA SUL
    PAVIMENTO
240 DATA 0,OCCHIO,PER TERRA C'È UN
    GIOIELLO A FORMA DI OCCHIO
250 DATA 22,LUME, VEDI UN LUME
260 DATA 0,ISPETTORE DELLE TASSE,È
    APPARSO UN ISPETTORE DELLE TASSE!
```

```
160 REM **PREP. MATRICI DEGLI OGGETTI**
170 READ NB
180 DIM OB(NB),OB$(NB),SI$(NB)
190 FOR I=1 TO NB:READ OB(I),OB$(I),SI$
    (I):NEXT
200 DATA 7,4,SACCHETTO,C'È UN
    SACCHETTO DI BIGLIE
210 DATA 14,MATTONE,C'È UN MATTONE AI
    TUOI PIEDI
220 DATA 24,CATENA,UNA CATENA PENDE
    DAL SOFFITTO
230 DATA 0,PISTOLA,C'È UNA PISTOLA SUL
    PAVIMENTO
240 DATA 0,OCCHIO,PER TERRA C'È UN
```

```
   GIOIELLO A FORMA□□□DI OCCHIO
250 DATA 22,LUME,VEDI UN LUME
260 DATA 0,ISPETTORE DELLE TASSE,È
   APPARSO UN ISPETTORE DELLE TASSE!
```

Ciascuna delle linee da 200 a 260 contiene tre elementi delle DATA relativi allo stesso oggetto. La 200 comprende anche un altro elemento supplementare: la cifra 7 all'inizio delle DATA comunica al computer il numero di elementi inseriti nelle DATA.

Il primo dato (7 nel nostro caso), viene adoperato per dimensionare le tre matrici (linea 180): OB contiene il numero dell'ambiente relativo a ogni oggetto, oppure il valore 0 se l'oggetto non esiste ancora (ossia appartiene ancora alle cose da scoprire nel corso dell'avventura), o, infine, il valore −1 se l'oggetto è *trasportato* dal giocatore. OB$ contiene il nome di ciascun oggetto e SI$ la descrizione. La linea 190 riempie le matrici con il contenuto delle frasi DATA (dalla linea 200 alla 260). Ogni frase è organizzata con lo stesso criterio: numero dell'ambiente, nome dell'oggetto e descrizione.

Questa routine può essere impiegata anche in altre avventure: non occorre apportare altre modifiche alla struttura, se non al primo elemento della DATA, che indica il numero degli elementi e serve per dimensionare sia il ciclo FOR ... NEXT che le matrici.

DISPOSIZIONE DEGLI OGGETTI

Adesso il programma possiede tutte le informazioni sulla natura e la collocazione degli oggetti. La seguente routine fa comparire la descrizione dell'oggetto nell'ambiente corrispondente:

```
360 REM **VISUALIZZAZIONE DEGLI OGGETTI
   NEL LUOGO APPROPRIATO**
370 FOR I=1 TO NB: IF B(I)=L THEN
   PRINT S$(I)
380 NEXT I
```

```
360 REM **VISUALIZZAZIONE DEGLI OGGETTI
   NEL LUOGO APPROPRIATO**
370 FOR I=1 TO NB:IF OB(I)=L□THEN
   PRINT SI$(I)
380 NEXT
```

Ora occorre una modifica nelle linee 330 e 340: cambiare le GOTO 400 in GOTO 370. Le linee 370 e 380 ispezionano la matrice contenente la collocazione degli oggetti: se uno degli elementi corrisponde all'ambiente in cui ci si trova (variabile L), allora compare la descrizione dell'oggetto, dopo quella dell'ambiente. Questa routine si

può usare senza modifiche per altre avventure.

AGGIUNGERE I VERBI

Ora l'avventura ha qualche oggetto sparso per gli ambienti, ma ancora inutilizzabile, poiché la macchina non comprende che le direttive NORD, SUD, EST e OVEST. È frustrante non poter raccogliere l'ambito sacco di biglie o proteggersi dall'ispettore delle tasse!

Occorre quindi *insegnare* al computer una serie di parole, riguardanti lo scopo di ciascun oggetto.

Vedremo in seguito come comportarci di fronte all'immissione di una parola non prevista.

Dal momento che il programma tratta le parole delle direzioni come verbi, la matrice R$ è il posto giusto per i verbi che indicano cosa fare degli oggetti e la matrice R il posto giusto per i numeri associati a ciascuna azione.

Sono comunque necessarie alcune modifiche: la prima riguarda i limiti del ciclo FOR ... NEXT alla linea 130. Si rimpiazzi quest'ultima con:

```
130 FOR K=1 TO 19:READ R$(K), R(K):
    NEXT K
```

```
130 FOR K=1 TO 19:READ R$(K), R(K):NEXT
```

Poi si aggiungano le linee 140 e 145.

```
140 DATA"NUOTO",5,"VUOTO",6,"LUCE",7,
    "FINE",8,"ELENCO",9,"UCCIDO",10,
    "SPARO",10,"AIUTO",11
145 DATA"PRENDO",2,"RACCOLGO",2,
    "PORTO",2,"METTO",3,"LASCIO",3,
    "MOLLO",3,"TIRO",4
```

```
140 DATA NUOTO,5,VUOTO,6,LUCE,7,FINE,8,
    ELENCO,9,UCCIDO,10,SPARO,10,AIUTO,11
145 DATA PRENDO,2,RACCOLGO,2,PORTO,2,
    METTO,3,LASCIO,3,MOLLO,3,TIRO,4
```

Ad ogni verbo corrisponde un numero: se due (o più) verbi hanno un valore identico, identica sarà l'interpretazione da parte del computer. Questo sistema permette di ampliare il vocabolario delle azioni, pur senza complicare il programma: se sono previsti i verbi TENGO, PORTO, PRENDO (il cui significato è pressoché identico), il giocatore non è costretto a rammentarsi quale dei tre usare. Volendo aggiungere nuove parole alle frasi DATA, basta cambiare il ciclo FOR ... NEXT alla linea 130 e accodare le voci in più in fondo alla linea 145. In tal caso occorreranno anche altre modifiche, che verranno spiegate in un'altra lezione.

ALTRE ROUTINE DI RICERCA

Completato l'elenco dei verbi, nella sezione appena vista, servono ora alcune routine che associno a ciascun verbo un'azione ben precisa (ad esempio quella di trasportare un dato oggetto, dopo averlo "preso").

La subroutine che inizia alla linea 3010 (già trascitta precedentemente) definisce V$, N$ e I, ossia un numero estratto dalla matrice R.

Grazie a questa breve routine, la macchina è in grado di comprendere il significato del verbo immesso (corrispondente al valore di I), e di agire in conseguenza.

Siccome lo Spectrum non possiede l'istruzione ON ... GOTO, la rispettiva versione differisce leggermente dalle altre.

Esiste già una matrice G, che contiene i numeri di linea per le descrizioni degli ambienti: la cosa più semplice è aggiungervi i numeri di linea relativi alle nuove routine.

Ecco perché la linea 30 risulta essere:

```
30 FOR N=1 TO 4: FOR M=1 TO 11: READ
   G(M,N): NEXT M: NEXT N
```

Infatti, la frase DATA ad essa associata contiene tutti i numeri di linea necessari (per una spiegazione più estesa si veda a pagina 346):

```
70 DATA 1010,1150,1240,1310,1410,1460,
   1500,1360,1080,1550,3110
```

Ecco ora la routine per la scelta della routine giusta a seconda del valore di I:

```
500 REM **ROUTINE DI RICERCA**
510 IF I=0 THEN GOTO 520
520 PRINT"NON RICONOSCO□";V$:
    GOTO 370
```

Se la subroutine 'Controllo Direttiva', che inizia alla linea 3010, non trova un abbinamento per V$ in R$, I è posto a zero e la linea 510 fa apparire il messaggio "NON RICONOSCO". Per ogni altro valore di I, la linea 515 ricerca nella matrice G il numero di linea corretto ed esegue una GOTO.

```
500 REM **ROUTINE DI RICERCA**
505 IF I=0 THEN GOTO 520
510 ON I GOTO 1010,1150,1240,1310,1410,
    1460,1500,1080,1550,3110
520 PRINT:PRINT "NON RICONOSCO□";V$:
    GOTO 370
```

Ogni numero dopo l'ON ... GOTO alla linea 510 è l'inizio di una routine. Ogni valore di I corrisponde ad un verbo (o gruppo di verbi) diverso. Se, ad esempio, I=10 (che corrisponde a 'uccido'), va scelta la decima routine della serie, ossia quella che inizia alla linea 1550. Se la subroutine 'Controllo Direttiva', che inizia alla linea 3010, non trova un abbinamento per V$ in R$, I viene posto a zero, nel quale caso l'ON ... GOTO non ha nessun effetto e compare il messaggio della linea 520.

TROUBLESHOOTER

- Ci si assicuri che i tre elementi di DATA, relativi agli oggetti, vengano letti nella giusta matrice. Tentando di far leggere una stringa a una matrice numerica, si ottiene un messaggio d'errore, ma uno scambio tra nomi e descrizioni si noterebbe soltanto durante il gioco.
- Per evitare un problema analogo, attenzione a far corrispondere l'ordine degli elementi delle DATA a quello di lettura delle matrici. Il corretto ordine è: ambiente, nome, descrizione.
- Si esegua un RUN *di collaudo* dell'avventura, dopo aver immesso gli oggetti, per assicurarsi che compaiano negli ambienti giusti.
- Si controlli la disposizione degli oggetti sulla griglia per evitare di ometterli dal gioco.

PRENDERE GLI OGGETTI

La routine che corrisponde al caso I=1 (il giocatore ha indicato una direzione verso la quale muoversi) è già stata immessa e si trova alle linee 1010-1060. Se I=2, si è scelto di 'prendere' un oggetto: TENGO, PORTO o PRENDO. Con la seguente routine, il giocatore può tenere con sé ogni oggetto che si trovi in quell'ambiente:

```
1140 REM **ROUTINE "PRENDI"**
1150 FOR G=1 TO NB
1160 IF N$=B$(G, TO LEN N$) THEN GOTO
     1190
1170 NEXT G
1180 PRINT "NON CAPISCO□";N$:GOTO 330
1190 IF B(G)=-1 THEN PRINT "CE L'HAI
     GIÀ": GOTO 330
1200 IF B(G)<>L THEN PRINT "QUI NON
```

```
   C'È!":GOTO 330
1210 PRINT "OK" : LET B(G)=-1
1220 GOTO 330
```

```
1140 REM **ROUTINE "PRENDI"**
1150 FOR G=1 TO NB
1160 IF N$=LEFT$(OB$(G),LEN(N$)) THEN
   1190
1170 NEXT
1180 PRINT "□NON CAPISCO□";N$:
   GOTO 330
1190 IF OB(G)=-1 THEN PRINT "CE L'HAI
   GIÀ!":GOTO 330
1200 IF OB(G)<>L THEN PRINT "QUI NON
   C'È!":GOTO 330
1210 PRINT "▒OK π":OB(G)=-1
1220 GOTO 330
```

```
1140 REM **ROUTINE "PRENDI"**
1150 FOR G=1 TO NB
1160 IF INSTR(OB$(G),N$)=1 THEN GOTO
   1190
1170 NEXT
1180 PRINT"□NON CAPISCO□";N$:GOTO
   330
1190 IF OB(G)=-1 THEN PRINT "CE L'HAI
   GIÀ!":GOTO 330
1200 IF OB(G)<>L THEN PRINT "QUI NON
   C'È!":GOTO 330
1210 PRINT "OK":OB(G)=-1
1220 GOTO 330
```

Le linee 1150-1170 esaminano la matrice che contiene i nomi degli oggetti (B$ per lo Spectrum, OB$ per gli altri) alla ricerca di quello indicato dal giocatore. Se l'oggetto non è trovato, il programma passa alla linea 1180, visualizzando la scritta NON CAPISCO, seguita dal nome immesso. Nel caso, invece, che l'oggetto sia stato trovato, l'esecuzione passa alla linea 1190 e vengono fatti due controlli. Il primo è per stabilire se l'oggetto è già in possesso del giocatore (linea 1190) e in caso affermativo compare il messaggio CE L'HAI GIÀ.

Il secondo controllo riguarda invece la presenza dell'oggetto nell'ambiente attraversato (consultando nuovamente la matrice B o OB). In caso negativo, viene visualizzato il messaggio QUI NON C'È. Ovviamente, entrambi i messaggi possono essere adattati al tipo di avventura giocato. Se l'oggetto *non* è trasportato e si trova nello stesso luogo del giocatore, la linea 1280 dà via libera e il valore corrispondente all'oggetto nella matrice che indica la collocazione degli oggetti (B o OB) viene posto uguale a −1.

ABBANDONO DEGLI OGGETTI

La routine 'Lascia' agisce in modo analogo, ma con effetto opposto, permettendo al giocatore di abbandonare un oggetto:

```
1230 REM **ROUTINE "LASCIA"**
1240 FOR G=1 TO NB
1250 IF N$=B$(G, TO LEN N$) THEN GOTO
   1270
1260 NEXT G:PRINT"NON
   CAPISCO□";N$:GOTO 330
1270 IF B(G)<>-1 THEN PRINT "NON
   L'HAI CON TE!":GOTO 330
1280 PRINT "OK": LET B(G)=L
1290 GOTO 330
```

```
1230 REM **ROUTINE "LASCIA"**
1240 FOR G=1 TO NB
1250 IF N$=LEFT$(OB$(G),LEN(N$)) THEN
   1270
1260 NEXT:PRINT"NON CAPISCO□";N$:GOTO
   330
1270 IF OB(G)<>-1 THEN PRINT "NON
   L'HAI CON TE!":GOTO 330
1280 PRINT "OK":OB(G)=L
1290 GOTO 330
```

```
1230 REM **ROUTINE "LASCIA"**
1240 FOR G=1 TO NB
1250 IF INSTR(OB$(G),N$)=1 THEN 1270
1260 NEXT:PRINT"NON CAPISCO□";N$:
   GOTO 330
1270 IF OB(G)<>-1 THEN PRINT "NON
   L'HAI CON TE!":GOTO 330
1280 PRINT "OK":OB(G)=L
1290 GOTO 330
```

Di nuovo, il nome dell'oggetto viene cercato nella matrice contenente le descrizioni *brevi* (linee da 1240 a 1260). Se l'oggetto viene trovato, la linea 1270 verifica che il giocatore l'abbia con sé e, in caso negativo, appare il messaggio NON L'HAI CON TE.

Qualora, invece, l'oggetto sia trasportato, la linea 1280 dà conferma con un OK, e l'appropriato elemento della matrice O (oppure OB) viene posto uguale al valore di L (corrispondente all'ambiente attraversato).

ELENCO DEL BOTTINO

Al cacciatore di avventure smemorato è utile un elenco di tutti gli oggetti raccolti durante la ricerca. Ecco una routine con questo compito:

```
1070 REM **INVENTARIO**
1080 PRINT "HAI CON TE:□";:LET IN=0
1090 FOR G=1 TO NB
1100 IF B(G)=-1 THEN PRINT TAB
   10;B$(G): LET IN=IN+1
1110 NEXT G
1120 IF IN=0 THEN PRINT "UN BEL
   NIENTE!"
1130 GOTO 330
```

```
1070 REM **INVENTARIO**
1080 PRINT "◣HAI CON TE:π□";:IN=0
1090 FOR G=1 TO NB
1100 IF OB(G)=-1 THEN PRINT
   TAB(10)OB$(G):IN=IN+1
1110 NEXT
1120 IF IN=0 THEN PRINT "UN BEL
   NIENTE!"
1130 GOTO 330
```

```
1070 REM **INVENTARIO**
1080 PRINT "HAI CON TE:□";:LET IN=0
1090 FOR G=1 TO NB
1100 IF OB(G)=-1 THEN PRINT
   TAB(10)OB$(G):IN=IN+1
1110 NEXT
1120 IF IN=0 THEN PRINT "UN BEL
   NIENTE!"
1130 GOTO 330
```

La linea 1080 visualizza la scritta HAI CON TE, seguita dall'elenco degli oggetti. Il ciclo FOR ... NEXT passa in rassegna ciascun elemento della matrice con la collocazione degli oggetti, alla ricerca di tutti quelli il cui valore è −1, a indicare che l'oggetto è trasportato: il nome dell'oggetto viene estratto dall'apposita matrice con le descrizioni brevi e visualizzato; il contatore degli oggetti, IN, cresce di 1.

Se non si hanno oggetti, IN resta a zero e, anziché un elenco di oggetti, la linea 1120 visualizza 'UN BEL NIENTE!'.

Le routine 'prendo', 'lascio' e 'inventario' si possono usare così come sono anche in altre avventure, purché NB si sia definito in una precedente routine.

Adesso conviene memorizzare anche questa parte di programma su nastro: la prossima volta verranno aggiunte le routine finali che riguardano l'ispettore delle tasse, il mattone, il lume, la scoperta dell'occhio, il finale dell'avventura e l'istruzione che descrive l'oggetto della ricerca.

Se si esegue il programma a questo stadio, si vedrà che, mentre alcune parti girano, in altre succedono cose strane: ciò accade perché mancano ancora alcune routine e il programma tenta di eseguire linee che ancora non esistono.

Si può usare un sintetizzatore vocale con un'avventura

Certamente. Un'avventura diviene ancor più avvincente se il computer è programmato in modo da pronunciare i messaggi, le direzioni e le descrizioni degli oggetti, anziché visualizzarli su schermo. Si consulti il manuale del sintetizzatore vocale, per vedere come far parlare l'apparecchio e si sostituiscano le istruzioni PRINT. I sintetizzatori vocali saranno trattati in una prossima lezione.

COMPLETIAMO L'AVVENTURA

- UNA ROUTINE D'AIUTO
- TRATTARE CON L'ISPETTORE
- PROBLEMI CON IL MATTONE
- ACCENDERE IL LUME
- ISTRUZIONI

L'avventura di INPUT è quasi completa: restano soltanto quelle routine che rendono unica l'avventura, perché stabiliscono le sue caratteristiche peculiari

Ora che gli elementi principali sono stati inseriti nel gioco, è il momento di rifinire il programma. Dobbiamo ancora inserire rischi e avvertimenti e manca ancora un modo per uscire dall'avventura in caso di successo.

Vanno aggiunte, infine, le istruzioni su come affrontare il gioco.

Poiché molte di queste routine sono scritte per contenere particolari specifici soltanto di una avventura, la programmazione che segue non solo completa questa avventura, ma mostra anche, in termini generali, tutto ciò che questa fase dello sviluppo comporta.

Pertanto, queste routine vanno adattate prima di poterle applicare ad altre avventure. Nella successiva lezione, verrà fatto un confronto tra i principi generali e le idee da incorporare in particolare giochi d'avventura.

SERVE AIUTO?

In ogni buon gioco d'avventura, al giocatore è consentito chiedere qualche aiuto. Perché negargli qualche suggerimento?

In risposta alla parola AIUTO, facciamo visualizzare un messaggio sullo schermo. La scelta del momento adatto e del contenuto dei messaggi sono a discrezione del programmatore. Si può scegliere, se si crede, di non offrire alcun messaggio, di darne alcuni a doppio senso o, infine, di limitare l'aiuto soltanto a pochi ambienti.

Per scegliere cosa inserire, è bene tornare al progetto originario e riguardare la traccia dell'avventura.

Nell'avventura di INPUT sono vari i punti in cui varrebbe la pena inserire un breve messaggio per il giocatore. Uno potrebbe essere un avvertimento riguardante la stanza buia, in modo che una richiesta di AIUTO in un ambiente adiacen-

te provochi un messaggio del tipo: ATTENTO PRIMA DI SCEGLIERE, o ancora più enigmatico.

Un altro punto adatto potrebbe essere in prossimità del fiume: se si decide di attraversare, si può anneggare secondo se si abbia o no il mattone con sé

Sono molti altri i luoghi in cui può servire un aiuto, ma supponiamo qui di non voler essere troppo generosi e di limitarsi al messaggio di fronte al fiume. Esso andrà messo in relazione con il numero dell'ambiente, il 7, e con la variabile che prende atto della presenza del mattone, OB (2):

```
3100 REM **ROUTINE DI "AIUTO"**
3110 IF L<>7 OR B(2)<>-1 THEN
    PRINT "SPIACENTE, QUI NON TI POSSO
    AIUTARE!": GOTO 330
3120 PRINT "I MATTONI PESANO E TI
    AFFATICANO. □□□LE BRACCIA TI
    FANNO MALE": GOTO 330
```

```
3100 REM **ROUTINE DI "AIUTO"**
3110 IF L<>7 OR OB(2)<>-1 THEN
    PRINT "SPIACENTE, QUI NON TI POSSO
    AIUTARE!":GOTO 330
3120 PRINT "I MATTONI PESANO E TI
    AFFATICANO. □□□LE BRACCIA TI
    FANNO MALE":GOTO 330
```

```
3100 REM **ROUTINE DI "AIUTO"**
3110 IF L<>7 OR OB(2)<>-1 THEN
    PRINT "SPIACENTE, QUI NON TI POSSO
    AIUTARE!":GOTO 330
3120 PRINT "I MATTONI PESANO E TI
    AFFATICANO. □□□LE BRACCIA TI
    FANNO MALE":GOTO 330
```

Se il giocatore non è sul fiume (L<>7) o non ha con sé il mattone (OB (2) <> -1 o B (2) <> -1), la linea 3110 visualizza il messaggio SPIACENTE, QUI NON TI POSSO AIUTARE! Se invece si è di fronte al fiume con il mattone e si chiede aiuto, la linea 3120 fa comparire: I MATTONI PESANO E TI AFFATICANO. LE BRACCIA TI FANNO MALE.

Volendo, si può aggiungere un intero elenco di condizioni e di relativi messaggi.

L'ISPETTORE DELLE TASSE

La parte del nemico, nella nostra avventura, la fa l'ispettore delle tasse, il cui scopo è recuperare un po' di crediti del giocatore, confiscandogli uno degli oggetti che questi porta con sé. Non fa molto caso a quel che arraffa e talvota si accontenta anche di un pagamento in mattoni!

Se il giocatore, quando l'ispettore fa le sue richieste, non ha la fortuna di avere qualcosa con sé, viene sbattuto in prigione e il gioco termina.

La presenza dell'ispettore delle tasse serve per inserire un elemento casuale imprevedibile, non legato a un ambiente o ad altre condizioni. Per le sue *apparizioni* si adopera la funzione RND, immancabile quando si tratta di generare numeri casuali nei programmi. Un sistema consiste nell'usare lo stesso metodo impiegato per gli oggetti, ma assegnando a caso la collocazione dell'acerrimo nemico. Ecco le linee di programma necessarie per le "improvvisate":

```
320 IF RND<(1/15) AND TA=0 THEN LET
    B(7)=L:LET TA=1
480 IF B(7)=L AND I<>10 THEN GOTO
    1590
```

```
320 IF INT(RND(1)*15+1)=1 AND TA=0
    THEN OB(7)=L:TA=1
480 IF OB(7)=L AND I<>10 THEN 1590
```

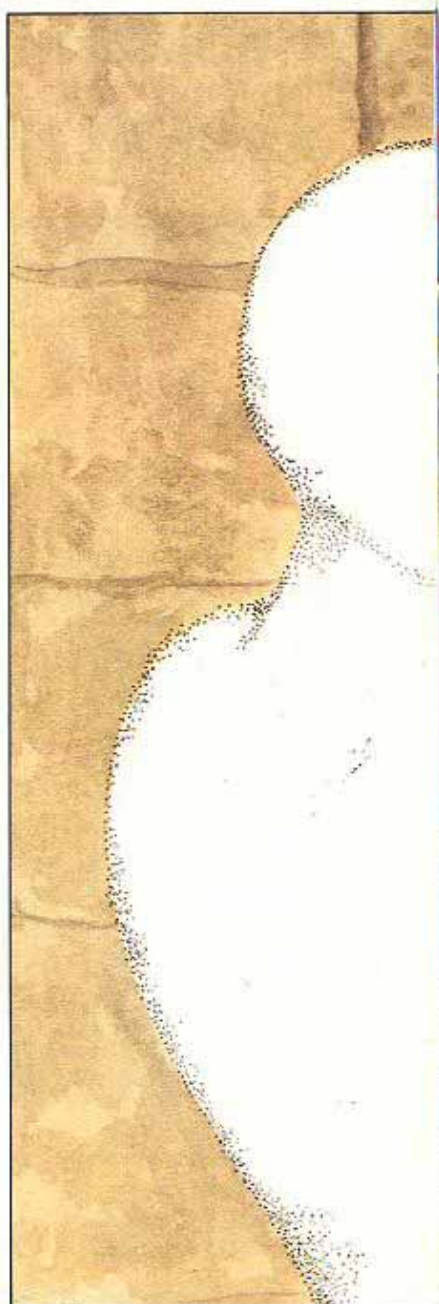
```
320 IF RND(15)=1 AND TA=0 THEN OB(7)
    =L:TA=1
480 IF OB(7)=L AND I<>10 THEN1590
```

```
320 IFRND(15)=1 AND TA=0 THENOB(7)
    =L:TA=1
480 IF OB(7)=L AND I<>10 THEN 1590
```

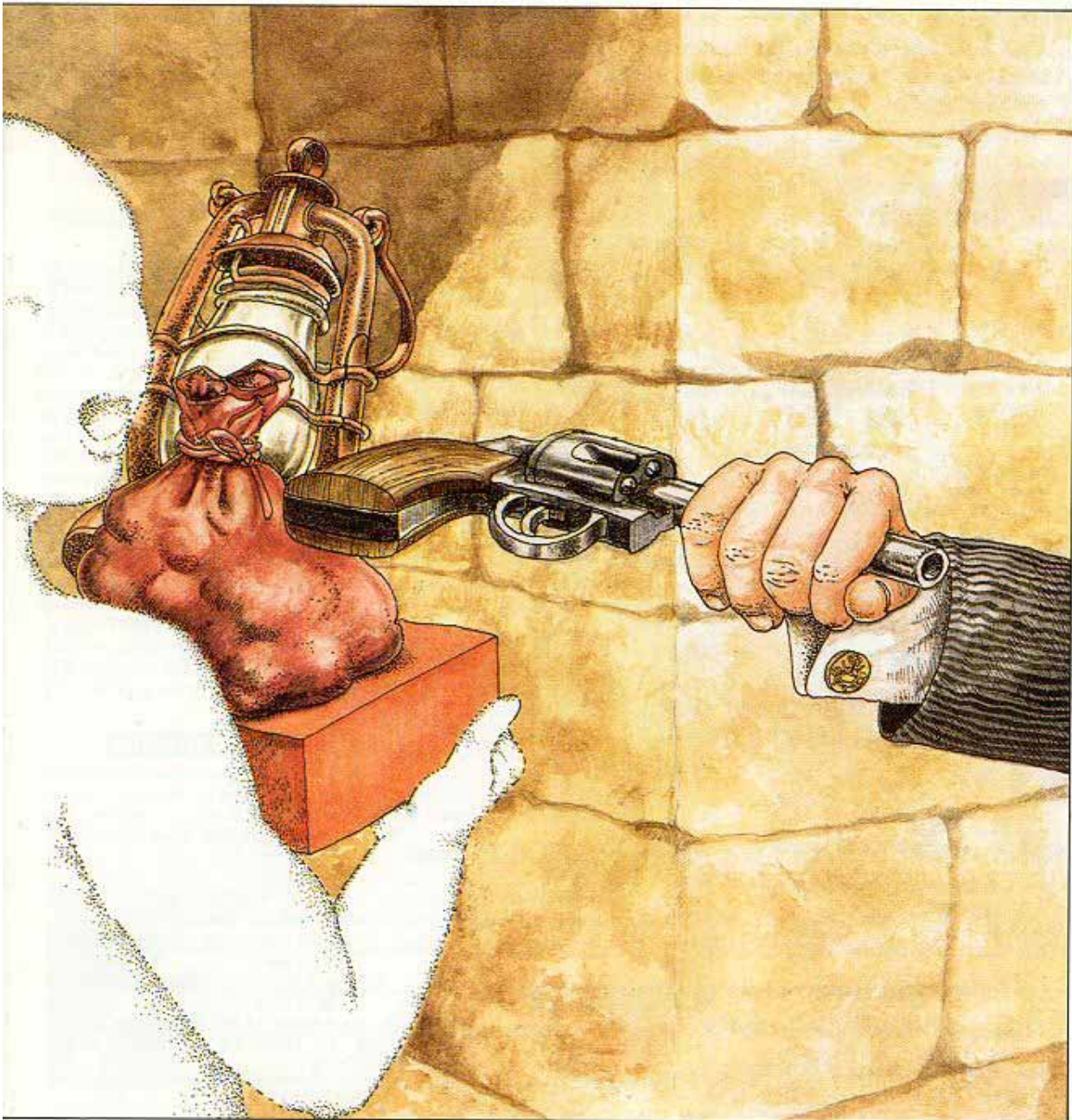
Inizialmente, Acorn e Spectrum pongono tutte queste variabili a zero.

La linea 320, in tutte le versioni, dà al giocatore una probabilità su 15 di incontrare l'ispettore (ecco il perché del 15, abbinato alla RND) e a quest'ultimo è concessa una sola apparizione nel corso di un gioco (verificata tramite la variabile TA).

Se il numero a caso è 1, (o minore di 1/15 sullo Spectrum) e l'ispettore non è ancora comparso, la linea 320 corregge il valore che corrisponde all'ispettore nella matrice relativa alla collocazione degli oggetti. Il messaggio dell'ispettore, contenuto nella matrice delle descrizioni, compare in modo analogo a quello usato per gli oggetti.

La linea 480 riguarda l'uccisione dell'ispettore e controlla se si è provato ad ucciderlo: in caso negativo, il programma va alla linea 1590.

TASSE: LA REAZIONE DEL GIOCATORE

Quando l'ispettore delle tasse ci minaccia, non c'è altra soluzione (stiamo parlando del gioco, ovviamente!) che spararagli con la pistola che si trova oltre il fiume:

```
1540 REM **ROUTINE "SPARA"**
```

```
1550 IF B(4) < > −1 THEN PRINT "E CON
    CHE COSA???":GOTO 320
1560 IF B(7) < > L THEN PRINT V$;"□A
    CHI?":GOTO 320
1570 PRINT B$(7);"□UCCISO!": LET B(7)
    = 0:GOTO 330
```

```
1540 REM **ROUTINE "SPARA"**
1550 IF OB(4) < > −1 THEN PRINT "E CON
    CHE COSA???":GOTO 320
1560 IF OB(7) < > L THEN PRINT V$;"□A
    CHI?":GOTO 320
1570 PRINT OB$(7);"□UCCISO!":OB(7) = 0:
    GOTO 330
```

La routine è richiamata quando il giocatore sceglie le parole UCCIDO o SPARO.

Se non si ha la pistola (OB (4) < > −1 o B (4) < > −1), la linea 1550 visualizza il

messaggio E CON COSA? Analogamente, se l'ispettore non è presente e il giocatore sceglie lo SPARO, la linea 156 chiede A CHI? La linea 1570 ci informa, con un laconico UCCISO!, sulla sorte del povero ispettore e corregge la matrice relativa alla colorazione degli oggetti in modo che l'ispettore non esista più.

LA VENDETTA DELL'ISPETTORE

Esiste però il risvolto della medaglia:

```
1580 REM **ISPETTORE DELLE TASSE**
1590 LET IN=0: LET B(7)=0
1600 FOR K=1 TO NB
1610 IF B(K)=-1 THEN LET IN=IN+1
1620 NEXT K
1630 IF IN=0 THEN PRINT "SICCOME NON
   HAI NIENTE CON TE□□VIENI RINCHIUSO
   IN UNA SPERDUTA CELLA BUIA": GOTO
   1360
1640 LET K=INT (RND*NB)+1: IF B(K)<
   >-1 THEN GOTO 1640
1650 PRINT "EGLI TI PORTA VIA IL□";B$(K):
   LET B(K)=0: GOTO 400
```

```
1580 REM **ISPETTORE DELLE TASSE**
1590 IN=0:OB(7)=0
1600 FOR K=1 TO NB
1610 IF OB(K)=-1 THEN IN=IN+1
1620 NEXT
1630 IF IN<>0 THEN 1640
1635 PRINT "□SICCOME NON HAI NIENTE
   CON TE,"
1638 PRINT "□VIENI RINCHIUSO IN UNA
   SPERDUTA CELLA BUIA":GOTO 1360
1640 K=INT(RND(1)*NB+1):IFOB(K)<>
   -1 THEN 1640
1650 PRINT "EGLI TI PORTA VIA
   IL□";OB$(K):OB(K)=0:GOTO 330
```

```
1580 REM **ISPETTORE DELLE TASSE**
1590 IN=0:OB(7)=0
1600 FOR K=1 TO NB
1610 IF OB(K)=-1 THEN IN=IN+1
1620 NEXT
1630 IF IN=0 THEN PRINT"SICCOME NON
   HAI NIENTE CON TE□□VIENI RINCHIUSO
   IN UNA SPERDUTA CELLA BUIA":GOTO
   1360
1640 K=RND(NB):IF OB(K)<>-1 THEN
   1640
1650 PRINT "EGLI TI PORTA VIA
   IL□";OB$(K):OB(K)=0:GOTO 330
```

```
1580 REM **ISPETTORE DELLE TASSE**
1590 IN=0:OB(7)=0
1600 FOR K=1 TO NB
1610 IF OB(K)=-1 THEN IN=IN+1
1620 NEXT
1630 IF IN=0 THEN PRINT"□SICCOME NON
   HAI NIENTE CON TE□□VIENI RINCHIUSO
   IN UNA SPERDUTA CELLA BUIA":GOTO
   1360
1640 K=RND(NB):IF OB(K)<>-1 THEN
   1640
1650 PRINT"EGLI TI PORTA VIA
   IL□";OB$(K):OB(K)=0:GOTO 330
```

L'ispettore può comparire solo una volta nel corso dell'avventura, cosicché la linea 1590 corregge la matrice relativa alla collocazione degli oggetti in modo che, per quel che riguarda il programma, l'ispettore non esista. Tuttavia, ciò non influenza questa routine, ma provvede a non far comparire l'ispettore neppure nell'ambiente successivo. IN è un contatore degli oggetti trasportati dal giocatore.

Le linee 1600-1620 esaminano la matrice contenente la collocazione degli oggetti, per controllare quali siano *in mano* al giocatore: per ogni oggetto trasportato, IN aumenta di 1.

Se non si hanno oggetti, il valore di IN resta zero e al giocatore viene servito un SICCOME NON HAI NIENTE CON TE, VIENI RINCHIUSO IN UNA SPERDUTA CELLA BUIA. Il gioco termina, saltando alla linea 1360, che chiede al giocatore se vuole ricominciare.

Se invece, si ha qualche oggetto con sé, la linea 1640 ne sceglie uno a caso, e lo "confisca".

Avvenuta la confisca, la linea 1650 informa il giocatore su ciò che gli è stato tolto dall'ispettore delle tasse e la matrice con la collocazione degli oggetti viene corretta in modo che tale oggetto non esista più.

FACCIAMO UN TUFFO

Questa è la routine richiamata, quando si decide di attraversare il fiume a nuoto:

```
1400 REM **ROUTINE "NUOTO"**
1410 IF L<>7 THEN PRINT "IN
   COSA?!!":GOTO 400
1420 IF B(2)=-1 THEN PRINT "CHE
   DISASTRO, SEI AFFOGATO!!":GOTO 1360
1430 IF B(4)>-1 THEN PRINT "HAI
   TROVATO UNA PISTOLA": LET B(4)=
   -1: GOTO 400
1440 PRINT "SEI TUTTO BAGNATO":GOTO
   400
```

```
1400 REM **ROUTINE "NUOTO"**
1410 IF L<>7 THEN PRINT"□IN COSA?!!":
   GOTO330
1420 IF OB(2)=-1 THEN PRINT "□CHE
   DISASTRO, SEI AFFOGATO!!":GOTO 1360
1430 IF OB(4)>-1 THEN PRINT"□HAI
   TROVATO UNA PISTOLA":LET OB(4)=
   -1:GOTO 330
1440 PRINT "□SEI TUTTO BAGNATO":GOTO
   330
```

La linea 1410 controlla se il giocatore è di fronte al fiume, altrimenti compare la domanda IN COSA? Siccome non esistono piscine o oceani nell'avventura, non ha senso includere una routine che si occupi di eventuali risposte: invece di un altro messaggio, ricompaiono semplicemente le direzioni da prendere. Tentando di attraversare il fiume, senza mollare il mattone, siamo spacciati: CHE DISASTRO, SEI AFFOGATO! Comunque, ci possiamo più o meno 'reincarnare', venendoci richiesto se vogliamo ripartire da capo.

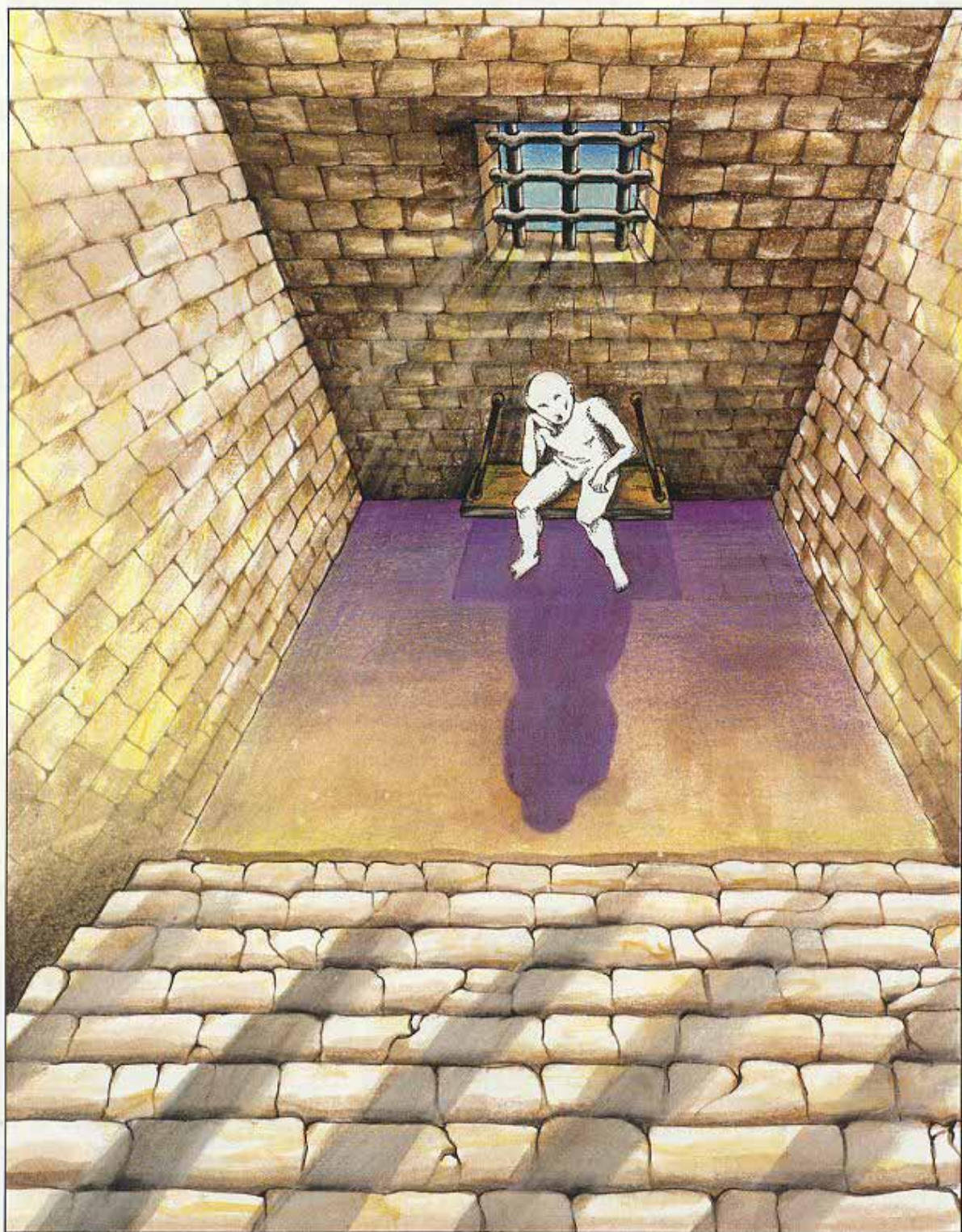
La linea 1430 verifica se il giocatore ha con sé una pistola, in caso contrario viene corretta la matrice relativa alla collocazione degli oggetti e compare il messaggio HAI TROVATO UNA PISTOLA.

Se, invece, il giocatore ha già trovato la pistola e sta di nuovo attraversando il fiume, la linea 1440 si limita a un SEI TUTTO BAGNATO!

INFINE, L'OCCHIO DELL'ICONA

Il giocatore può recuperare il leggendario occhio soltanto se ha ritrovato il sacchetto delle biglie: basta rovesciarlo ed appare la pietra preziosa. Ecco la routine:

```
1450 REM **ROUTINE "VUOTO"**
1460 IF N$<>"SACCHETTO"(TO LEN N$)
   THEN PRINT "NON PUOI VUOTARLO!":
   GOTO 400
1470 IF B(1)<>-1 THEN LET G=1:
   GOTO 1270
1480 PRINT "LE BIGLIE SONO ROTOLATE
   SUL PAVIMENTO": LET B(5)=L: GOTO
   370
```

```
1450 REM **ROUTINE "VUOTO"**
1460 IN=0:IF N$=LEFT$("SACCHETTO",
   LEN(N$)) THEN IN=1
1465 IF IN<>1 THEN PRINT "□NON PUOI
   VUOTARLO!":GOTO 330
1470 IF OB(1)<>-1 THEN G=1:GOTO
   1270
1480 PRINT "□LE BIGLIE SONO ROTOLATE
   SUL PAVIMENTO":OB(5)=L: GOTO 370
```

```
1450 REM **ROUTINE "VUOTO"**
1460 IN=INSTR("SACCHETTO",N$):IF IN<
    >1 THEN PRINT"□NON PUOI
    VUOTARLO!":GOTO 330
1470 IF OB(1)<>-1 THEN G=1:GOTO
    1270
1480 PRINT"□LE BIGLIE SONO ROTOLATE
    SUL PAVIMENTO":OB(5)=L: GOTO 370
```

Questa viene richiamata quando il giocatore dà l'istruzione 'VUOTA *qualcosa*'. La linea 1460 controlla che quel *qualcosa* sia il sacchetto, altrimenti (N$ <> "SACCHETTO") e appare il messaggio NON PUOI VUOTARLO!. La linea 1470 verifica la presenza del sacco (OB (1) <> -1 o B (1) <> -1): se non c'è, invece di un nuovo messaggio, salta alla linea 1270 per visualizzare il messaggio NON CE L'HAI, visto prima. Se il sacchetto c'è, allora il programma passa alla linea 1480 informandoci che LE BIGLIE SON ROTOLATE SUL PAVIMENTO. La matrice con la collocazione degli oggetti viene corretta di conseguenza, dato che l'occhio si trova ora nell'ambiente attraversato.

Non c'è bisogno di un messaggio, perché, saltando alla linea 370, si può sfruttare la descrizione lunga dell'oggetto, già immessa con la frase DATA (si veda alla linea 240).

ACCENDERE IL LUME

Per *vedere* le uscite dalla stanza buia, occorre accendere il lume: senza di esso, si rimane bloccati nell'oscurità impenetrabile. La routine per accendere il lume è la seguente:

```
1490 REM **ROUTINE "LUME"**
1500 IF N$<>"LUME" (TO LEN N$) THEN
    PRINT "NON PUOI FARLO": GOTO 400
1510 IF B(6)<>-1 THEN LET G=6:
    GOTO 1270
1520 IF LA=1 THEN PRINT "È GIÀ
    ACCESO": GOTO 400
1530 LET LA=1: LET DA=0: PRINT
    "OK":GOTO 330
```

```
1490 REM ** ROUTINE "LUME" **
1500 IN=0:IF N$=LEFT$("LUME",LEN
    (N$)) THEN IN=1
1505 IF IN<>1 THEN PRINT "□NON PUOI
    FARLO":GOTO 330
1510 IF OB(6)<>-1 THEN G=6:GOTO
    1270
1520 IF LA=1 THENPRINT "È GIÀ
    ACCESO":GOTO 330
1530 LA=1:PRINT "OK":GOTO 330
```

```
1490 REM ** ROUTINE "LUME"**
1500 IN=INSTR("LUME",N$):IF IN<>1
    THENPRINT"□NON PUOI FARLO":GOTO
    330
1510 IF OB(6)<>-1 THENG=6:GOTO
    1270
1520 IF LA=1 THEN PRINT"È GIÀ
    ACCESO":GOTO330
1530 LA=1:PRINT"OK":GOTO 330
```

Essa viene richiamata qualora il giocatore digiti 'ILLUMINO *qualcosa*'. La linea 1500 (la 1505, nel Commodore) è molto simile alla corrispondente linea nella routine 'VUOTO'; essa controlla se è stato indicato il lume: come in quella occasione, compare un messaggio analogo, NON PUOI FARLO!, qualora il lume non sia trasportato.

La linea 1530 pone il flag 'lume acceso' a 1, poi visualizza un OK di conferma.

LA FINE È VICINA

Dal soffitto della sala del trono, dove finalmente è giunto il giocatore, pende una catena. Che fare? Tirarla oppure no?

Ecco la routine che 'amministra' le conseguenze:

```
1300 REM **ROUTINE "TIRO"**
1310 IF N$="CATENA"(TO LEN N$) THEN
    LET IN=1:IF IN=1 AND L<>24 THEN
    PRINT "NON ACCADE NULLA": GOTO 400
1320 IF IN<>1 THEN PRINT "NON PUOI
    TIRARLO!": GOTO 400
1330 IF B(5)<>-1 THEN PRINT
    "SPLASH...GLOB...GLOB... SEI FINITO
    NELLA FOGNA!!!": GOTO 1360
1335 REM **FINE DELL'AVVENTURA**
1340 PRINT "COMPLIMENTI!!! HAI
    COMPLETATO L'AVVENTURA"
1360 PRINT "'ANCORA UNA PARTITA
    (S/N)?"
1370 LET A$=INKEY$: IF A$<>"S" AND
    A$<>"N" THEN GOTO 1370
1380 IF A$="S" THEN RUN
1390 STOP
```

```
1300 REM ** ROUTINE "TIRO" **
1310 IN=0:IF N$=LEFT$("CATENA",LEN
    (N$))THEN IN=1
1315 IF IN=1 AND L<>24 THEN PRINT
    "NON ACCADE NULLA":GOTO 330
1320 IF IN<>1 THEN PRINT "□NON PUOI
    TIRARLO!":GOTO330
1330 IF OB(5)=-1 THEN 1340
1335 PRINT "□SPLASH...GLOB...GLOB... SEI
    FINITO NELLA FOGNA!!!":GOTO 1360
1340 REM ** FINE DELL'AVVENTURA **
1350 PRINT "COMPLIMENTI!!! HAI
    COMPLETATO L'AVVENTURA"
1360 PRINT:PRINT "▓ANCORA UNA
    PARTITA (S/N)?"
1370 GET A$:IF A$<>"S" AND A$<
    >"N" THEN 1370
1380 IF A$="S" THEN RUN
1390 PRINT "□□":POKE 53280, 14:END
```

sul Vic 20 si cambi la linea 1390 in:

```
1390 PRINT "□□": POKE 36879,27:END
```

```
1300 REM **ROUTINE "TIRO"**
1310 IN=INSTR("CATENA",N$):IF IN=1
    AND L<>24 THEN PRINT "□NON
    ACCADE NULLA":GOTO 330
1320 IF N<>1 THEN PRINT"□NON PUOI
    TIRARLO!":GOTO330
1330 IF OB(5)<>-1 THEN PRINT
    "□SPLASH...GLOB...GLOB...SEI FINITO
    NELLA FOGNA!!!":GOTO 1360
1340 REM **FINE DELL'AVVENTURA**
1350 PRINT"COMPLIMENTI!!! HAI
    COMPLETATO L'AVVENTURA"
1360 PRINT:PRINT "□ANCORA UNA
    PARTITA (S/N)?"
1370 A$=INKEY$:IF A$<>"S" AND A$<
    >"N" THEN 1370
1380 IF A$="S" THEN RUN
1390 END
```

I possessori di Acorn, alla linea 1370, devono usare GET$ anziché INKEY$.

La linea 1310 riguarda la possibilità che il giocatore chiami in causa la catena fuori della sala del trono senza tirarla e fa comparire il messaggio NON SUCCEDE NIENTE.

Se il giocatore tenta di tirare altri oggetti dell'avventura, gli viene comminato un semplice NON PUOI TIRARLO! (linea 1320).

Dopo, accade l'imprevisto: se il giocatore è nella sala del trono, ma non ha trovato l'occhio, si visualizza SPLASH ... GLOB ... GLOB ... SEI FINITO NELLA FOGNA!!!, giungendo a una ben misera conclusione dell'avventura!

Al contrario, se il giocatore ha recuperato l'occhio e tira la catena nella sala del trono, viene inneggiato un gratificante COMPLIMENTI!! HAI COMPLETATO L'AVVENTURA.

Infine, viene data la possibilità di ricominciare con una nuova partita: le linee 1360-1380, che si occupano di ciò, in pratica servono soltanto se il giocatore è finito in prigione o nella fogna.

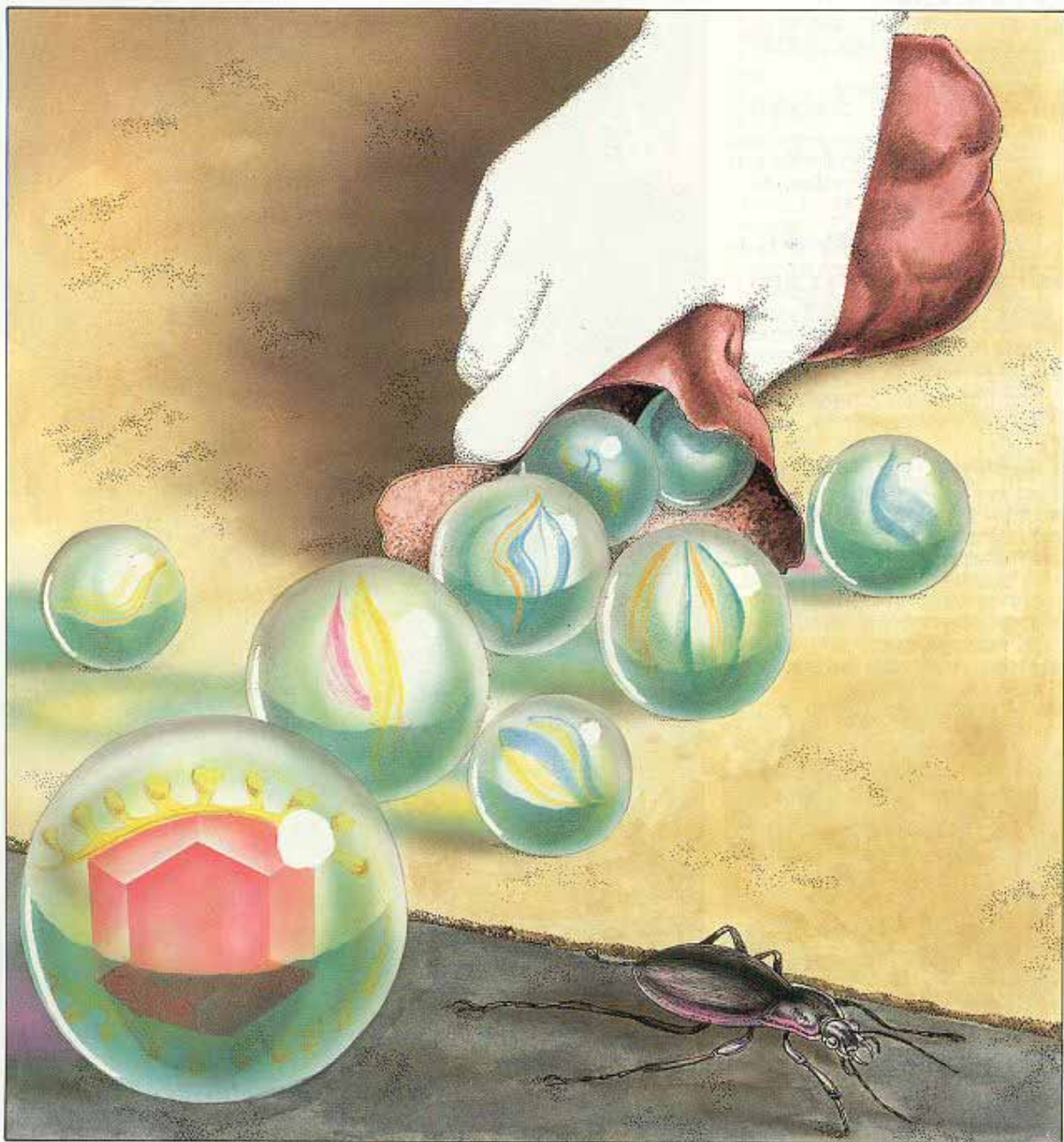
LE ISTRUZIONI

Il gioco è quasi pronto. Non manca che un tocco finale: le istruzioni.

Senza istruzioni, non si conoscono né lo scopo di tutti gli sforzi né cosa fare durante il gioco. Prima di aggiungere la routine per le istruzioni, si faccia un controllo di quanta memoria rimane (si veda a pagina 268). Se ne resta poca, è meglio togliere tutte le linee REM (rammentandosi di rinumerare tutte le GOSUB che indirizzano il programma ad esse, per evitare errori!).

Occorre valutare attentamente la quantità di istruzioni da dare. In relazione alla memoria disponibile, si deciderà quanti suggerimenti dare a questo o a quel livello, tenendo anche conto di aspetti quali il formato dello schermo sul proprio apparecchio (se bastano le righe oppure no) e via dicendo.

L'avventura di INPUT è molto sempli-

ce, quindi la routine delle istruzioni è breve e contiene poche informazioni:

```
80 CLS:PRINT "VUOI VEDERE LE ISTRUZIONI
   (S/N)?"
90 LET A$=INKEY$:IF A$="" THEN GOTO
   90
95 IF A$="S" THEN GOSUB 6000
6000 REM **ISTRUZIONI**
6010 CLS:PRINT:PRINT:"□□A SEGUITO DI
   UN DISSESTO FINANZIARIO□□□□SEI
   FUGGITO DAL
   TUO□□□□□□□□□PAESE"
6020 PRINT:PRINT "□□IL RITROVAMENTO
   DEL PREZIOSO
   OCCHIO□□□□PURPUREO DI UNA
   FAVOLOSA ICONA□□E IL
   SUPERAMENTO DI UNA
   PROVA□□□MOLTO
   PERICOLOSA□□□PORREBBE FINE AI
   TUOI PROBLEMI"
6030 PRINT:PRINT "□□□EVITA A TUTTI I
   COSTI L'ISPETTORE DELLE TASSE"
6040 PRINT AT 20,3;"PREMI QUALSIASI
   TASTO PER CONTINUARE"
6050 LET A$=INKEY$:IF A$="" THEN
   GOTO 6050
6060 RETURN
```

```
10 PRINT "VUOI LE ISTRUZIONI (S/N)?"
20 GET A$:IF A$="" THEN 20
30 IF A$="S" THEN GOSUB 6000
6000 REM**ISTRUZIONI**
6010 PRINT:PRINT "□□A SEGUITO DI
   UN DISSESTO FINANZIARIO SEI FUGGITO
   DAL TUO PAESE"
6020 PRINT:PRINT"□□IL RITROVAMENTO
   DEL PREZIOSO OCCHIO"
6025 PRINT "PURPUREO DI UNA FAVOLOSA
   ICONA E IL SUPERAMENTO"
6027 PRINT"DI UNA PROVA MOLTO
   PERICOLOSA"
6029 PRINT "PORREBBE FINE AI TUOI
   PROBLEMI"
6030 PRINT:PRINT"□□EVITA A TUTTI I
   COSTI L'ISPETTORE DELLE TASSE"
6040 PRINT TAB(8) "PREMI QUALSIASI
   TASTO PER CONTINUARE"
6050 GET A$:IF A$="" THEN 6050
6060 RETURN
```

```
1 CLS:PRINT "VUOI LE ISTRUZIONI
   (S/N)?"
20 A$=GET$
30 IF A$="S" THEN GOSUB 6000
6000 REM**ISTRUZIONI**
6010 CLS:PRINT' "□□A SEGUITO DI UN
   DISSESTO FINANZIARIO SEI FUGGITO DAL
   TUO PAESE"
6020 PRINT' "□□IL RITROVAMENTO DEL
```

```
   PREZIOSO OCCHIO PURPUREO DI UNA
   FAVOLOSA ICONA E IL SUPERAMENTO DI
   UNA PROVA MOLTO PERICOLOSA
   PORREBBE FINE AI TUOI PROBLEMI"
6030 PRINT:PRINT "□□EVITA A TUTTI I
   COSTI L'ISPETTORE DELLE TASSE"
6040 PRINT TAB(8,23) "PREMI QUALSIASI
   TASTO PER CONTINUARE"
6050 A$=GET$
6060 RETURN
```

```
10 CLS:PRINT "VUOI VEDERE LE ISTRUZIONI
(S/N)?"
20 A$=INKEY$:IF A$="" THEN 20
30 IF A$="S" THEN GOSUB 6000
6000 REM**ISTRUZIONI**
6010 CLS:PRINT:PRINT"□□A SEGUITO DI
   UN DISSESTO FINANZIARIO SEI
   □□□□□□□□FUGGITO DAL TUO
   PAESE"
6020 PRINT:PRINT "□□IL RITROVAMENTO
   DEL PREZIOSO OCCHIO
   PURPUREO□□□DI UNA FAVOLOSA
   ICONA E IL SUPERAMENTO□□□DI UNA
   PROVA MOLTO PERICOLOSA
   PORREBBE□□□FINE AI TUOI
   PROBLEMI"
6030 PRINT:PRINT "□□EVITA A TUTTI I
   COSTI L'ISPETTORE□□□□□DELLE
   TASSE"
6040 PRINT@451, "PREMI QUALSIASI
   TASTO PER CONTINUARE"
6050 A$=INKEY$:IF A$="" THEN 6050
6060 RETURN
```

Adesso, non resta che passare alla SAVE.

La prossima lezione riguarda l'uso della struttura sviluppata per "Il Prezioso Occhio dell'Icona Purpurea", quale base per altre divertenti avventure.

Microtip

Ravvivare l'avventura

Una volta completata la stesura del programma di gioco, ci si può sbizzarrire a inserire, ad esempio, una pagina iniziale con i titoli e una finale con le congratulazioni. La gamma di arricchimenti possibili è legata a quanta memoria rimane (si veda a pagina 268), ma quanto meno si dovrebbero aggiungere un po' di colori. La pagina col titolo d'apertura dovrebbe attirare l'occhio, mentre quella conclusiva, con le congratulazioni, dovrebbe sottolineare il merito di aver conseguito un tale successo, ricorrendo a un po' di grafica o a una breve melodia.

AVVENTURE: IL PROSSIMO PASSO

I giochi d'avventura possono diventare una vera "febbre": se ne vorrebbero sempre di nuovi. Ecco come partire dall'avventura di INPUT per inventare le proprie

A questo punto, il gioco d'avventura di INPUT dovrebbe essere ormai registrato e pronto all'uso. Durante lo sviluppo, abbiamo visto come mettere assieme tutti gli elementi che lo compongono, a partire dalla prima, rudimentale traccia della trama. Inoltre, sono già stati forniti alcuni suggerimenti su eventuali varianti. Stavolta, vedremo in modo più complessivo come usare il gioco come base per creare le nostre avventure.

Poiché molte delle modifiche dipendono dalla fisionomia generale della nuova avventura, non è possibile descriverle molto dettagliatamente; tuttavia non sarà difficile intervenire, seguendo attentamente i consigli qui riportati. Se alcune delle tecniche descritte dovessero inizialmente intimorire, basterà applicarle gradualmente scrivendo qualche prima avventura molto semplice, per afferrarne a fondo i principi.

Gli sviluppi qui presentati, sotto il profilo della programmazione non introducono concetti nuovi. Pertanto, se qualche sezione del programma BASIC non risultasse chiara, si consultino le precedenti sezioni di Programmazione Basic e si troveranno opportune spiegazioni.

Purtroppo, i possessori di Vic 20 e di Spectrum con 16K non possono ampliare molto l'avventura, a causa della limitata memoria. Ciò non significa, però, che ogni estensione sia fuori portata o che non si possano scrivere avventure con più ambienti. Ovviamente, un maggior sviluppo di alcune parti del programma andrà a detrimento di altre: è una questione di preferenze personali.

TEMI PER LA PROPRIA AVVENTURA

Prima di scrivere la propria avventura, occorre una trama adatta.

Di per sé, la struttura di una buona storia d'avventura è di solito molto semplice: c'è un inizio, uno sviluppo centrale e un finale. Il tutto poggia sulla sequenza dei problemi da affrontare, impostata dal programmatore. Fortunatamente, non occorre la penna di Agata Christie per scrivere giochi d'avventura: le fonti di ispirazione, come abbiamo già visto, sono innumerevoli. Comunque, per facilitare i primi tentativi, ecco alcuni suggerimenti.

La struttura dell'avventura potrebbe incentrarsi sulla ricerca di un *colpevole*. Ad esempio, la scena iniziale potrebbe raffigurare una stanza in cui si vede il cadavere di un uomo pugnalato e lo scopo del gioco potrebbe essere trovare l'assassino. Al posto del nostro "ispettore delle tasse", potremmo inserire un maggiordomo, col compito di aiutare o di ostacolare il giocatore.

Oppure, potremmo usare in vari modi il classico personaggio del *naufrago*: in modo tradizionale, tipico delle storie di pirati, oppure più attuale (il solo superstite di un disastro aereo). Ambientando la storia nel futuro, il protagonista, membro di una spedizione galattica, si perde in prossimità di un pianeta ostile, lontano dalla nostra civiltà. Lo scopo dell'avventura può essere quello di trovare un modo per fuggire, nonostante l'astronave sia in avaria. Gli indigeni potrebbero costituire "il nemico" e gli spunti per creare imprevedibili vie di fuga sarebbero molti.

Altre trame di fuga potrebbero riguardare Alcatraz o un Super Carcere immaginario. Ci si potrebbe ispirare alla trama di qualche libro, magari disegnando una cartina dei luoghi interessati, utile per un più realistico progetto degli ambienti.

Le spie, tradizionali o industriali (a caccia di un progetto per un nuovo supercomputer?), sono un'altra grande fonte di ispirazione.

Perché non ispirarsi alla Storia? Le Crociate sono un tipico scenario per giochi d'avventura, ma anche qualsiasi altra guerra può servire da sfondo.

Infine, si potrebbe concepire un'avventura ambientata nel "Dopobomba nucleare", corredata di esseri mutanti, di bande di predatori affamati alla ricerca di cibo e acqua non contaminata e così via.

ALTRI AMBIENTI

L'avventura di INPUT è volutamente molto contenuta: redigendo le proprie, si vedrà il programma crescere rapidamente, ampliando le limitate possibilità iniziali.

Seguendo le istruzioni alle pagine 296-301, si potrà creare una griglia adeguata ai propri programmi. La griglia di INPUT è di 6×4 ambienti (24 in tutto), ma ne vengono usati soltanto 12. Che si decida o meno di utilizzare la medesima griglia, si può modificare il programma esistente (meno lavoro, ma più confusione) oppure scriverne uno completamente nuovo (maggior lavoro, ma più linearità). A seconda del criterio scelto, si richiamerà il programma salvato su nastro o, se si possiede una stampante, lo si listerà su carta. Gli adattamenti dipendono dalla dimensione della griglia necessaria. Con 24 ambienti o meno, si può usare la griglia esistente e riportarvi i nuovi ambienti. Se occorre una griglia più grande, la si disegni e poi la si numeri come fatto in precedenza.

Chiarito quale griglia usare, si può immettere la propria serie di descrizioni degli ambienti, per rimpiazzare quelle esistenti, dalla linea 5000 in poi.

Come nel programma originale, ogni descrizione deve essere seguita dalla linea contenente le uscite possibili. Le variabili N,S,E e W corrispondono a nord, sud, est e ovest e possono essere poste a 0 oppure a 1: 0 indica che in quella direzione non c'è uscita, 1 che si può uscire. Scrivere opportune linee REM, con i numeri degli ambienti, è una fatica che vale la pena di fare.

Poi vanno cambiati i numeri di linea nelle ON ... GOSUB (linee 330-350): il primo numero dopo GOSUB alla linea 330 è il numero della linea corrisponde alla descrizione dell'ambiente 1. Se nell'avventura non c'è un ambiente 1 (non è detto che si usino tutti i quadrati della griglia), al suo posto si immetta 0. Il secondo numero corrisponde alla linea della seconda descrizione e così via. Si rammenti che deve esserci un numero per ciascun ambiente della griglia.

GLI SPOSTAMENTI

Se si è progettata una griglia di grandezza diversa da quella usata per l'avventura di INPUT, occorre cambiare la routine del movimento (dalla linea 1000 alla 1040). Più specificamente, andranno cambiate le linee del nord e del sud, la 1010 e la 1030, se la griglia non è più larga 6 quadrati, poiché per cambiare fila sulla griglia si aggiunge o si sottrae 6. Basta contare di quanti quadrati è la larghezza della griglia e sostituire quel numero al 6.

GLI OGGETTI

Gli oggetti della nuova avventura saranno diversi da quelli dell'avventura di INPUT, per cui saranno necessari molti cambiamenti alle linee 160-260.

Si conti il numero di oggetti da usare nell'avventura: questo numero dà il valore di NB e sarà il primo elemento della frase DATA, alla linea 200, usato in seguito per DIMensionare le matrici alla linea 180 e predisporre il ciclo FOR ... NEXT nel programma.

Il listato del programma risulta più chiaro se ogni linea racchiude la descrizione di un solo oggetto, ma se gli oggetti sono numerosi (o se occorre risparmiare memoria) converrà inserire più descrizioni in ogni linea. Qualsiasi criterio si adotti per le DATA, l'ordine deve essere corretto, poiché ciascuno dei tre elementi di ogni gruppo di dati è destinato a un matrice diversa. Il giusto ordine è: numero dell'ambiente, nome e descrizione. Se l'oggetto appare soltanto in un secondo momento dell'avventura (forse solo dopo che il giocatore l'ha trovato), oppure è qualcosa che appare a caso (come il nostro ispettore delle tasse), il numero dell'ambiente deve essere zero.

PAROLE NUOVE

È bene redigere un elenco di tutte le direttive a disposizione del giocatore nel corso dell'avventura. Nell'elenco andranno incluse le parole singole (come le direzioni, AIUTO, INVENTARIO, ecc.) e i comandi di due parole (quali PRENDO LUME, UCCIDO DRAGO, ecc.).

Questo secondo gruppo di direttive va poi attribuito a V$ oppure a N$, le variabili dei verbi e dei nomi, anche se non sempre ci si atterrà alla definizione grammaticale. Infatti, ciò che interessa sono le parole singole e la prima parola di ogni coppia. Per il programma queste sono i verbi, V$. I verbi vanno raggruppati insieme secondo i significati: MANGIO e INGOIO, per esempio, possono equivalere. A ciascuno di questi gruppi viene attribuito un numero: non ha importanza quale, purché si sappia quale numero si riferisce a quel particolare gruppo di parole.

Ora le modifiche: la routine dei verbi va dalla linea 11Ø alla 15Ø. I verbi e i numeri ad essi associati sono immessi nelle frasi DATA alle linee 14Ø e 15Ø, disposti a coppie (il "verbo", seguito dal numero).

Si DIMensionino opportunamente le matrici alla linea 12Ø e si modifichi adeguatamente il ciclo FOR ... NEXT alla linea 13Ø, in base al numero totale di verbi usati.

LE ROUTINE DEI VERBI

Ogni singolo gruppo di verbi (caratterizzato da un particolare numero) richiede una specifica routine.

È difficile fornire istruzioni esplicite su come scrivere queste routine, perché esse dipendono troppo dalle particolari "azioni" cui esse si riferiscono, nel contesto di una particolare avventura.

Tuttavia, esistono alcune routine che sono adatte più o meno a tutte le avventure, come quelle associate ai verbi PRENDO o LASCIO. In particolare, la direttiva INVENTARIO (alle linee 107Ø-113Ø) è la stessa in ogni avventura e la routine non va cambiata, purché la matrice sia la stessa e NB, il numero degli oggetti, abbia lo stesso significato anche nella nuova avventura. Un'altra routine, facilmente "riciclabile", è quella dell'accensione del lume (linee 149Ø-153Ø).

Le restanti routine, forse, non hanno un valore sufficientemente generico, ma possono costituire una traccia per la stesura di quelle nuove. La loro funzione fondamentale è quella di verificare se il giocatore tenta di adoperare l'oggetto giusto nel giusto ambiente. Se il posto è sbagliato, il programma visualizza un messaggio adatto alla situazione, del tipo NON QUI. Ad ogni buon conto, il giocatore deve essere sempre informato sull'effetto delle direttive impartite. In altre parole, qualsiasi direttiva immessa deve esser seguita da una risposta sullo schermo.

Una volta messe a punto, le routine dei verbi vanno incorporate nel programma e, se questo è numerato come l'avventura di INPUT, la loro collocazione sarà tra le linee 107Ø e 2999. Il computer deve saper scegliere la routine giusta in accordo col verbo usato dal giocatore; per ottenere ciò occorre modificare la linea 15Ø: è sufficiente esaminare il proprio elenco di verbi e, seguendo l'ordine numerico, immettere le linee corrispondenti alle relative routine, dopo l'ON ... GOTO.

ROUTINE D'AIUTO

L'ultima routine alla quale prestare attenzione è quella di AIUTO: si valuti in quale ambiente e a quali condizioni si intende fornire assistenza al giocatore, e si adoperi a tal scopo una linea IF ... THEN.

Una diversa impostazione dell'avventura comporterà anche altri piccoli interventi, ad esempio l'eliminazione della linea che fà apparire l'ispettore (la 32Ø).

LE VARIABILI USATE

Per facilitare gli interventi sul programma, ecco un elenco delle variabili e delle matrici, assieme alla loro funzione.

R$()	matrice contenente verbi e risposte
R()	matrice dei numeri delle risposte

Gli elementi corrispondenti, in queste due matrici, si riferiscono ad uno stesso oggetto.

OB()	matrice contenente il numero dell'ambiente di ciascun oggetto
OB$()	matrice contenente i nomi degli oggetti
SI$()	matrice contente le descrizioni degli oggetti

Gli elementi corrispondenti, in queste tre matrici, si riferiscono ad uno stesso oggetto.

NB	il numero di oggetti dell'avventura; usato per DIMensionare le matrici e per predisporre i cicli FOR ... NEXT
L	ambiente attraversato dal giocatore
LA	flag sullo stato del lume: 1 = acceso e Ø = spento
TA	flag dell'ispettore delle tasse
N, E, S, W	direzioni d'uscita: 1 se c'è un'uscita e Ø se non c'è
I$	immissione del giocatore, prima di esser suddivisa in verbi e nomi
V$	la parte verbale di I$
N$	la parte nominale di I$
I	numero corrispondente a un particolar significato del verbo
IN	numero di oggetti nell'INVENTARIO
A$	la risposta a VUOI RICOMINCIARE?
G	numero dell'oggetto abbandonato, elemento G della matrice OB

I programmi per lo Spectrum sono un po' diversi dagli altri perché il BASIC dello Spectrum non possiede ON ... GOTO né ON ... GOSUB. Non sono però mancanze gravi: il programma gira ugualmente bene e ciò non ostacola eventuali ampliamenti.

DESCRIZIONI DEGLI AMBIENTI

Nella modifica della versione Spectrum, il primo passo è l'immissione di tutte le descrizioni degli ambienti, relative alla griglia adottata.

Si mettano le descrizioni in fondo al programma, come nell'avventura di INPUT. Occorre rispettare l'ordine richiesto nella fase di lettura: ogni descrizione deve essere seguita dalla linea contenente l'informazione sulla direzione d'uscita. Le variabili N, S, E e W si riferiscono a nord, sud, est e ovest e i valori Ø e 1 indicano, rispettivamente, l'assenza o la presenza di un'uscita.

Non si dimentichi l'uso di opportune linee REM, che facilitano l'associazione tra i numeri e le descrizioni degli ambienti.

PAROLE NUOVE

Si faccia un elenco di tutte le direttive che il giocatore potrà dare al computer nel corso dell'avventura.

In questa fase sono importanti i *verbi* che, per quanto concerne il gioco, non sono ristretti alla rigorosa definizione grammaticale, ma si riferiscono alla prima di due parole immesse dal giocatore o anche a singoli vocaboli. Questi *verbi* vanno raggruppati assieme, in base al significato, cosicché parole quali PRENDO e RACCOLGO abbiano lo stesso effetto sullo svolgimento dell'avventura.

A ciascun gruppo va associato un numero distintivo.

I verbi, seguiti dai rispettivi numeri, vanno inseriti nelle frasi DATA alle linee 14Ø e 15Ø. Ricordiamo che le stringhe vanno racchiuse tra virgolette.

Occorre poi riDIMensionare le matrici

Nel progettare una nuova avventura si deve costantemente pensare a chi giocherà con essa, ai suoi interessi e a ciò che conosce, oltre che alla sua intelligenza: il gioco non va condito con indovinelli che richiedano una laurea in *sociologia astrofisica*, per esempio. Se nel gioco vengono posti quesiti su dati precisi (il raggio della Terra, per esempio), ci si assicuri che la risposta con la quale confrontiamo quella del giocatore sia effettivamente esatta, consultando, se necessario, un libro di testo specifico.

alla linea 120 e regolare il ciclo FOR ... NEXT alla linea 130, in base al numero totale di verbi usati.

LE ROUTINE DEI VERBI

Ogni gruppo di verbi richiede una specifica routine.

Siccome tali routine dipendono dalle particolari caratteristiche di un'avventura, è difficile poter fornire indicazioni esplicite su come scriverle. Tuttavia, alcune delle routine dell'avventura di INPUT sono comodamente "riciclabili".

Per esempio, le routine PRENDO e LASCIO possono essere ricopiate pari pari in qualsiasi avventura, perché il loro impiego è fondamentale per usare gli oggetti del gioco. Anche quella relativa all'INVENTARIO non richiede cambiamenti: si può scegliere se inserirla o meno, ma sarebbe strana un'avventura senza di essa.

Le altre routine, invece, dipendono troppo dalla trama dell'avventura, ma se ne può utilizzare la struttura per impostare quelle nuove. Occorre valutare dove e con che cosa adoperare un verbo.

Si verifichi che la direttiva venga adoperata nell'ambiente giusto e con l'oggetto giusto. Si cerchi di prevedere eventuali risposte errate, creando apposite routine di "risposta" ad esse.

Una volta scritte, tutte queste routine vanno incorporate tra le linee 1390 e 2999 del programma. Il computer deve essere in grado di scegliere, tra le routine, quella connessa al verbo usato dal giocatore.

LA MATRICE G

Nella versione per lo Spectrum, i numeri di linea relativi alle descrizioni degli ambienti e alle routine dei verbi sono contenute nella matrice G.

Il passo successivo, nella redazione del nuovo programma, consiste quindi nell'immettere tutti questi numeri in G: si vedano le linee da 40 a 70. Le prime tre si riferiscono alle descrizioni degli ambienti, l'ultima contiene tutti i numeri di linea relativi alle routine dei verbi.

Le DIMensioni della matrice G dipendono dal numero di elementi nella frase DATA più lunga e dal numero di frasi DATA presenti. A questi due valori corrispondono, rispettivamente, il primo e il secondo indice della matrice.

Si immettano poi tutte le frasi DATA nelle linee dalla 40 alla 70 (se queste non dovessero bastare, si creino nuove linee con una numerazione intermedia). Poiché è un fatto veramente raro che il numero degli elementi sia uguale per tutte le frasi

DATA, occorre inserire un numero sufficiente di zeri, per uniformarle tutte alla frase DATA con più elementi.

Le linee da 330 a 350 scelgono la corretta descrizione dell'ambiente, selezionando il numero di linea, ossia il giusto elemento nella matrice G. Il secondo indice corrisponde alla riga della matrice in cui si trova il numero, per cui occorre assicurarsi della correttezza di tale indice in ogni linea GOTO, specie se si sono aggiunte nuove linee di DATA.

GLI SPOSTAMENTI

Se la griglia della nuova avventura è di dimensioni diverse da quella usata nell'avventura di INPUT, va modificata la routine di movimento alle linee 1000-1040.

Più precisamente, vanno modificate le linee relative al nord e al sud (1010 e 1030), se la griglia non è più larga sei quadrati. Basta contare il numero di quadrati della larghezza della nuova griglia e sostituire questo numero al 6.

GLI OGGETTI

I cambiamenti più ampi riguardano le linee da 160 a 260, poiché gli oggetti nella nuova avventura non saranno certamente gli stessi dell'avventura di INPUT.

Si contino quanti oggetti ci sono e la lunghezza sia del nome più lungo che della descrizione più lunga. Il numero di oggetti è il primo elemento delle DATA alla linea 200 e servirà come indice per DIMensionare le matrici alla linea 180 per regolare cicli FOR ... NEXT del programma. Il secondo indice nella matrice B$ è la lunghezza del nome più lungo e il secondo indice in S$ è la lunghezza della descrizione più lunga.

Il listato risulta più chiaro se viene usata una linea di DATA per ciascuno degli oggetti, ma se questi ultimi sono molto numerosi sarà necessario raggrupparne più d'uno per linea. Qualunque sia il criterio adottato, l'ordine degli elementi nelle DATA deve essere corretto, poiché i tre elementi di ciascun gruppo di DATA sono destinati a matrice diversa. Il giusto ordine è: numero dell'ambiente, nome e descrizione. Se un oggetto compare solo in un secondo momento (forse dopo che il giocatore lo ha scovato), oppure compare a caso (come il nostro ispettore delle tasse), il numero dell'ambiente deve essere zero.

ROUTINE D'AIUTO

L'ultima routine cui dedicare attenzione è quella d'AIUTO: si valuti dove si intende fornire un suggerimento al giocatore e si usi a tal scopo una linea IF ... THEN.

Altri interventi possono implicare la modifica o l'eliminazione di alcune linee di programma, come ad esempio quella per far apparire l'ispettore, la 320.

USO DELLE VARIABILI

Per facilitare gli interventi sul programma, ecco un elenco delle variabili e delle matrici, assieme alla loro funzione.

G() matrice contenente i numeri di linea delle descrizioni degli ambienti e delle routine dei verbi

R$() matrice contenente verbi e risposte

R() matrice contenente i numeri dei verbi

Gli elementi corrispondenti, nelle ultime due matrici, si riferiscono a coppie di verbi/significati.

OB() matrice contenente il numero d'ambiente di ogni oggetto

OB$() matrice contenente i nomi degli oggetti

S$() matrice contenente le descrizioni degli oggetti

Gli elementi corrispondenti, in queste tre matrici, si riferiscono allo stesso oggetto.

NB il numero degli ogetti nella avventura. Usato per DIMensionare le matrici e per predisporre i cicli FOR ... NEXT

L ambiente attraversato dal giocatore

LA flag sullo stato del lume: 1 acceso e 0 spento

N, S, E, W direzioni di uscita: 1 se c'è un'uscita e 0 se non c'è

I$ l'immissione per intero, prima di esser divisa in verbi e nomi

V$ la parte verbale di I$

N$ la parte nominale di I$

I numero associato a un particolare significato del verbo: serve per controllare se un particolare verbo è stato usato in determinate fasi del programma

IN il numero di oggetti nell'INVENTARIO

A$ la risposta a VUOI RICOMINCIARE?

G il numero dell'oggetto abbandonato, elemento della matrice B